Application Report
SPRACK4–September 2019

AM335x EMIF tools

Catalog Processors

ABSTRACT

This document provides detailed steps outlining the procedure to initialize Texas Instruments' processors to access external DDR memories using the accompanying tools included in the AM335x EMIF Tools application.

Contents

List of Tables

Trademarks

1 Introduction

At the center of every application is the need for memory. With limited on-chip processor memory, external memory serves as a solution for large software systems and data storage, and an unstable external memory interface can result in system failures or hinder software development. To prevent potential system level anomalies and ensure robust systems, hardware must be configured correctly and tested thoroughly.

The AM335x EMIF Tools application focuses on post layout activities, including configuring the Texas Instrument processors for accessing external double data rate (DDR) memories, optimizing delay locked loop (DLL) ratios to compensate for board routing skew. This application report provides a detailed description on how to use the associated application files. The document overview provides a complete list of processors and memory types supported by the AM335x EMIF Tools application. The spreadsheet discussed in this application report can be downloaded from this link.

Supported Features of AM335x EMIF Tools

The AM335x EMIF Tools application supports the following features:

- TI SOCs: AM335x, AMIC110
- DDR Types: DDR3/DDR3L
- Basic EMIF configuration topics, including:
  - Initializing the EMIF and DDR for basic read/write functionality
  - Compensating for signal skew

2 EMIF Configuration

The following section describes how to use the supporting application files to configure the EMIF controller for DDR memory accesses.

2.1 Preliminary Requirements

Before using the supporting application files, you must have access to the following system application information:

- The data sheet of the selected DDR memory
- PCB trace lengths of the DDR clock (DDR_CK) and strobe signals (DDR_DQS) (required for deriving the right settings for leveling)

2.2 Generating EMIF Register Values

To help define the EMIF configuration register values, the AM335x EMIF Tools application provides an EMIF register configuration workbook. The workbook is divided into several worksheets, and requires specific information pertaining to the system application environment.

Table 1. EMIF Register Configuration Worksheets

Worksheet	Description
Step1: System Details	User Input: System information
Step2: DDR Timings	User Input: timing information from DDR data sheet
Step3: Board Details	User Input: board trace lengths of clock and DQS
EMIF Tool	Output: register value calculation
Registers	Output: final register values
GEL	Output: GEL formatted #define to be used in GEL scripts
u-boot	Output: u-boot structures and function call to be used in board.c

2.2.1 Step1: System Details

This worksheet requires you to input both high-level system application details, as well as specific I/O settings for the DDR pins of the TI application processor and DDR memory.

Step 1A seeks system level details:

- Board name: String is used in u-boot structure names.
- DDR memory type: Only DDR3 is supported in this tool. This cell cannot be modified.
- DDR memory frequency: This is the frequency of the DDR clock.
- DDR data bus width per EMIF: Only a 16-bit data bus is supported. This cell cannot be modified.
- DDR DQS: Only differential DQS is supported. This cell cannot be modified.

Step 1B DDR memory specifications:

- Speed Bin: Data Rate: Input the data rate that corresponds with the speed grade of the DDR3 IC being purchased. For example, if purchasing a DDR3-1866 device, then put 1866 in this field. This is independent of the speed at which the bus is actually running, which is covered in Step 1A. This value is used to determine the JEDEC values defined in the following worksheets.
- Density (per device): Input the density of a single DDR device.
- Number of rows: Enter the number of row bits of the DDR memory.
- Number of columns: Enter the number of column bits of the DDR memory.
- Number of banks: Enter the number of banks of the DDR memory.
- Speed bin: CAS latency: Enter the CAS latency at the max rated speed of the device, not the operating speed. The value does not correspond to the actual operational CAS latency. The value is used to determine the JEDEC values defined in the following worksheets.
- Width (per device): Enter the bus width of a single DDR device.
- Average periodic refresh interval: Enter the refresh rate of the DDR device.

Step 1C DDR memory I/O settings:

TI recommends using the settings in the “TI recommendation” column for these inputs.

- ODT/Rtt_Nom: This value applies to the on-die termination of the DDR memory I/O pins.
- Dynamic ODT/Rtt_Wr: This value applies to on-die termination during DDR writes when the dynamic ODT mode is enabled.
- Output driver impedance: This value applies to the output driver impedance of the DDR memory I/O pins.

Step 1D) ZQ Calibration settings:

Stay within the recommendations found in the “TI recommendations” column.

- ZQCL on self refresh: Enable (1) or disable (0) ZQ calibration on self-refresh.
- Temperature sensitivity: Input the max temperature sensitivity of the DDR device.
- Voltage sensitivity: Input the maximum voltage sensitivity of the DDR device.
- Temperature drift rate: Input the maximum temperature drift rate of the system.
- Voltage drift rate: Input the maximum voltage drift rate of the system.

Step 1E EMIF controller I/O settings:

TI recommends using the settings in the “TI recommendation” column for these inputs.

- ODT: Input the on-die termination of the EMIF I/O pins on the AM335x.
- Slew rate: Clk: Input the desired slew rate for the DDR clock signals driven by the AM335x.
- Slew rate: Addr/Ctrl: Input the desired slew rate for the DDR address and control signals driven by the AM335x.
- Slew rate: Data/Strobe: Input the desired slew rate for the DDR data and strobe signals driven by the AM335x.
- Output driver impedance: Clk : Input the desired output impedance for the AM335x DDR clock signals.

- Output driver impedance: Addr/Ctrl: Input the desired output impedance for the AM335x DDR address and control signals.
- Output driver impedance: DQS: Input the desired output impedance for the AM335x DDR DQS signals.
- Output driver impedance: Data: Input the desired output impedance for the AM335x DQ signals.

2.2.2 Step2: DDR Timings

This worksheet requires you to input DDR timing values that can be found in the DDR memory data sheet. Here are some tips in filling out this worksheet:

- DDR data sheets provide the timing parameters either in clock cycles (tCK) or ns, or sometimes the maximum of both. Fill out either or both parameters that are provided in the DDR data sheet. If a value doesn't exist in the DDR data sheet, leave it blank, or fill with zero. The Final Bit Value column calculates the value in tCK clock cycles. This value is used in calculating values in the Registers tab.
- Some cells are white. These are automatically calculated based on input from other values. Do not adjust these cells. Only input values in the yellow cells.
- Some values depend on the speed bin of the device. Use the values for the speed bin of the device you are using. For example, if using a DDR3-1866 device, use the speed bin table associated with DDR3-1866, and choose the values in the row associated with your maximum operating speed.
- Some values depend on the page size and device density. Choose the correct parameters when filling out the worksheet.
- In some cases, the spec is given only as tCK or ns. In that case, enter the number in the associated column and leave the other blank. In other cases, the spec might say something similar to "max(3nCK or 6ns)", in which case enter '3' under the tCK column and '6' under the ns column.
- All listed parameters require a minimum timing value, unless otherwise noted by '(max)'
- Ensure the values are correct when inputting parameters from the data sheet. There is a data validation check based on expected JEDEC parameters. If a cell is red, this means the values input do not meet the JEDEC timing standards. Correct these based on the DDR data sheet values.
- "Final Bit Field Values" are all "minus 1" values, that is, a value of 5 corresponds to 6 tCK. The JEDEC column represents the actual number of tCK periods.

2.2.3 Step3: Board Details

For this worksheet, you must have specific board design details.

Step 3A defines the signal delay per inch for the board, which can typically be kept at 180 ns per inch.

Step 3B: Input the trace lengths per byte for DDR CK and DQS. Because both signals are differential, you can take the average of DDR_CK and DDR_CKn (or DDR_DQS and DDR_DQSn) and input the length (in inches) for each byte in each cell. Include the total trace length from the AM335x to the memory, including via lengths. If using one DDR device, input the same length for both byte 0 and byte 1 for the clock signal. The results of the calculations can be found in steps 3C through 3E in the green cells. If you have a fly-by topology with 2 x8 DDR devices, these values are used in the software leveling algorithm described in Section 3. **If you only have one DDR device, you do not need to perform the software leveling algorithm.** The values are used in subsequent tabs (GEL and u-boot) to provide proper DDR PHY configuration.

2.2.4 Registers

After the previous worksheets have been completed, you can access the Registers worksheet to show the calculated values for each bit field. This worksheet is for information purposes only. Refer to this to find how individual bit fields are programmed based on the inputs provided. Further worksheets are provided which provide the full configuration needed for the controller and PHY registers.

2.2.5 GEL

The GEL tab provides the output to be used in a GEL configuration script for the DDR controller and PHY configuration. Use these #define lines to replace similar lines in the GEL you are using. This lets the GEL scripts configure the DDR controller and PHY for your specific board design. Only change the resulting lines. The rest of the GEL scripts should remain the same.

2.2.6 u-boot

The u-boot tab provides code fragments to be used in u-boot for DDR DPLL, controller, and PHY configuration. The structures should be placed in <u-boot>/board/ti/am335x/board.c to define the appropriate parameters for your board design. The config_ddr() function calls the DDR initialization routine with pointers to the DDR parameters for your design. The get_dpll_ddr_params() function should return a pointer to the DPLL_DDR parameters. It gets called by setup_dplls(), which subsequently uses that structure to configure DPLL_DDR.

2.3 Files Needed for This Procedure

Code Composer GEL file: This can be found in the zip file link at the beginning of this application note.

Executable CCS .out file: This can be found in the zip file link at the beginning of this application note.

3 Software Leveling

NOTE: Software leveling is only necessary if you have a fly-by topology with more than one DDR memory device. If you have a single 16-bit DDR device, the results from the EMIF tool spreadsheet represent the optimal values needed for the DDR PHY, and thus you do not need to perform the following steps.

1. Change the following section of the GEL file based on the values calculated in the AM335x EMIF Configuration Tool. Refer to the 'GEL' tab of the spreadsheet for the correct values. Copy the lines from the GEL tab and replace the similar lines in the GEL file you are using. Example is below.

```
//*****************************************************************
//DDR3 PHY parameters
//*****************************************************************

#define  CMD_PHY_CTRL_SLAVE_RATIO    0x00000100
#define  CMD_PHY_INVERT_CLKOUT       0x00000001

#define  DATA_PHY_RD_DQS_SLAVE_RATIO    0x00000038
#define  DATA_PHY_FIFO_WE_SLAVE_RATIO   0x000000FC
#define  DATA_PHY_WR_DQS_SLAVE_RATIO    0x00000078
#define  DATA_PHY_WR_DATA_SLAVE_RATIO   0x000000B8

#define  DDR_IOCTRL_VALUE           0x0000018B

//*****************************************************************
//EMIF parameters
//*****************************************************************
#define ALLOPP_DDR3_READ_LATENCY    8
#define ALLOPP_DDR3_SDRAM_TIMING1   0x0AAAE4DB
#define ALLOPP_DDR3_SDRAM_TIMING2   0x266B7FDA
#define ALLOPP_DDR3_SDRAM_TIMING3   0x501F867F

#define ALLOPP_DDR3_SDRAM_CONFIG    0x61C051B2

#define ALLOPP_DDR3_REF_CTRL        0x00000c30
#define ALLOPP_DDR3_ZQ_CONFIG       0x50074BE4
```

2. If your board includes VTT termination, ensure the VTT regulator is enabled when running the software leveling algorithm. The GEL file includes function to enable/disable the VTT regulator using a GPIO. If your regulator is GPIO controlled, ensure that the GPIO used by these functions matches the GPIO that is used on your custom board.
3. Connect the board to Code Composer and run the GEL file Scripts->AM335x System Initialization (this may run automatically on connect if it is part of the OnTargetConnect() GEL function). Load the DDR3_slave_ratio_search_auto_ver2.out file. After loading the file, ensure that the processor is in supervisor mode by looking for “SPV” in the lower right corner of the CCS window. If it shows “USR”, execute the script Script->default->AM335xStartState, to change back to “SPV” mode.
4. Run the code. The console window in CCS prompts you for some input values. Input each value from the 'Board Details' tab of the spreadsheet (input in hex with no leading '0x') corresponding to the prompt. The values are in sections 3C and 3E of the spreadsheet. The program may take time to run (especially if using XDS100), as it may iterate in a loop several times to come up to the optimal values. The console window shows progress and appears as in this example:

```
[CortxA8]
Enter the PHY_INVERT_CLKOUT value (0 or 1) from the spreadsheet
1

Enter the Seed RD_DQS_SLAVE_RATIO Value in Hex to search the RD DQS Ratio Window
40

Enter the Seed FIFO_WE_SLAVE_RATIO Value in Hex to search the RD DQS Gate Window
114

Enter the Seed WR_DQS_SLAVE_RATIO Write DQS Ratio Value in Hex to search the Write DQS Ratio
Window
87

************************************************************
        The Slave Ratio Search Program Values are...
************************************************************
PARAMETER                       MAX  |  MIN  | OPTIMUM |  RANGE
************************************************************
DATA_PHY_RD_DQS_SLAVE_RATIO    0x06a | 0x007 |  0x038  | 0x063
DATA_PHY_FIFO_WE_SLAVE_RATIO   0x1d3 | 0x04f |  0x111  | 0x184
DATA_PHY_WR_DQS_SLAVE_RATIO    0x0f9 | 0x00d |  0x083  | 0x0ec
DATA_PHY_WR_DATA_SLAVE_RATIO   0x139 | 0x04d |  0x0c3  | 0x0ec
************************************************************
rd_dqs_range = 38
fifo_we_range = 111
wr_dqs_range = 83
wr_data_range = 0

Optimal values not reached, rerunning program with new values...

************************************************************
        The Slave Ratio Search Program Values are...
************************************************************
PARAMETER                       MAX  |  MIN  | OPTIMUM |  RANGE
************************************************************
DATA_PHY_RD_DQS_SLAVE_RATIO    0x06b | 0x007 |  0x039  | 0x064
DATA_PHY_FIFO_WE_SLAVE_RATIO   0x1cd | 0x048 |  0x10a  | 0x185
DATA_PHY_WR_DQS_SLAVE_RATIO    0x0f9 | 0x00d |  0x083  | 0x0ec
DATA_PHY_WR_DATA_SLAVE_RATIO   0x139 | 0x04d |  0x0c3  | 0x0ec
************************************************************
rd_dqs_range = 1
fifo_we_range = 7
wr_dqs_range = 0
wr_data_range = 0

Optimal values not reached, rerunning program with new values...

************************************************************
        The Slave Ratio Search Program Values are...
```

```
************************************************************
PARAMETER                       MAX   |   MIN   | OPTIMUM |   RANGE
************************************************************
DATA_PHY_RD_DQS_SLAVE_RATIO    0x06b | 0x007 |  0x039  | 0x064
DATA_PHY_FIFO_WE_SLAVE_RATIO   0x1c7 | 0x047 |  0x107  | 0x180
DATA_PHY_WR_DQS_SLAVE_RATIO    0x0f9 | 0x00d |  0x083  | 0x0ec
DATA_PHY_WR_DATA_SLAVE_RATIO   0x139 | 0x04d |  0x0c3  | 0x0ec
************************************************************
rd_dqs_range = 0
fifo_we_range = 3
wr_dqs_range = 0
wr_data_range = 0

Optimal values not reached, rerunning program with new values...

************************************************************
        The Slave Ratio Search Program Values are...
************************************************************
PARAMETER                       MAX   |   MIN   | OPTIMUM |   RANGE
************************************************************
DATA_PHY_RD_DQS_SLAVE_RATIO    0x06b | 0x007 |  0x039  | 0x064
DATA_PHY_FIFO_WE_SLAVE_RATIO   0x1cb | 0x047 |  0x109  | 0x184
DATA_PHY_WR_DQS_SLAVE_RATIO    0x0f8 | 0x00d |  0x082  | 0x0eb
DATA_PHY_WR_DATA_SLAVE_RATIO   0x138 | 0x04d |  0x0c2  | 0x0eb
************************************************************
rd_dqs_range = 0
fifo_we_range = 2
wr_dqs_range = 1
wr_data_range = 0

Optimal values have been found!!

************************************************************
        The Slave Ratio Search Program Values are...
************************************************************
PARAMETER                       MAX   |   MIN   | OPTIMUM |   RANGE
************************************************************
DATA_PHY_RD_DQS_SLAVE_RATIO    0x06b | 0x007 |  0x039  | 0x064
DATA_PHY_FIFO_WE_SLAVE_RATIO   0x1cb | 0x047 |  0x109  | 0x184
DATA_PHY_WR_DQS_SLAVE_RATIO    0x0f8 | 0x00d |  0x082  | 0x0eb
DATA_PHY_WR_DATA_SLAVE_RATIO   0x138 | 0x04d |  0x0c2  | 0x0eb
************************************************************

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

5. When the program has finished running, use the 4 optimal values (in the OPTIMUM column) to program the appropriate DDR PHY slave ratio register values. Start by replacing the ones in the GEL, as shown in the following example. Save the GEL after changing the optimum values, then terminate the debug session and power cycle the board.

```
//*********************************************************************
//DDR3 PHY parameters
//*********************************************************************

#define  DATA_PHY_RD_DQS_SLAVE_RATIO    0x39  //replace with optimum value from SW leveling
algorithm
#define  DATA_PHY_FIFO_WE_SLAVE_RATIO   0x109 //replace with optimum value from SW leveling
algorithm
#define  DATA_PHY_WR_DQS_SLAVE_RATIO    0x82  //replace with optimum value from SW leveling
algorithm
#define  DATA_PHY_WR_DATA_SLAVE_RATIO  0xC2  //replace with optimum value from SW leveling
algorithm
```

6. Connect to the board again in CCS. The initialization script should automatically run. To test the results, open up a memory window in DDR space (0x80000000), and try to read/write values successfully. There are also DDR test scripts in the GEL which you can run (Scripts->AM335x DDR Tests). If the results look stable, you have finished. These values should be programmed during the

DDR initialization routine in the boot code (for example, in u-boot). The u-boot tab in the spreadsheet has a ddr_data structure which can be used to program the optimum values calculated by the algorithm. Replace the correct data structure elements with the appropriate values:

- .datardsratio0 is DATA_PHY_RD_DQS_SLAVE_RATIO value
- .datawdsratio0 is DATA_PHY_WR_DQS_SLAVE RATIO value
- .datafwsratio0 is DATA_PHY_FIFO_WE_SLAVE_RATIO value
- .datawrsratio0 is DATA_PHY_WR_DATA_SLAVE_RATIO value

4 Procedural Notes

Here are some notes on this procedure:

- If you rerun this program on the same board, you may get slightly different results. But these values could also be used. The program tries to find the proper timing window for reads and writes, and chooses the middle of that window to allow for maximum timing margin. thus, these optimal values may be a little different with each run of this test program.
- The program defines the following ranges to determine when to stop looking for optimal values. These values were chosen based on multiple runs with different boards TI tested.
  - define RD_DQS_OPTIMAL_RANGE 2
  - define WR_DQS_OPTIMAL_RANGE 3
  - define FIFO_WE_OPTIMAL_RANGE 2
  - define WR_DATA_OPTIMAL_RANGE 3
- If the program results in all zeros for values, then something has gone wrong and it was not able to converge on optimal values. Check the values obtained in the spreadsheet and the values input while running the program in CCS. Also, run the algorithm with the ARM processor in Supervisor mode (review Step 3 above).

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI’s products are provided subject to TI’s Terms of Sale (www.ti.com/legal/termsofsale.html) or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI’s provision of these resources does not expand or otherwise alter TI’s applicable warranties or warranty disclaimers for TI products.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265