

# KeyStone DDR3 Initialization

*High-Performance and Multicore Processors**Thomas Johnson*

## Abstract/Purpose/Benefit

The initialization of the DDR3 DRAM controller on KeyStone DSPs is straightforward as long as the proper steps are followed. However, if some steps are omitted or if some sequence sensitive steps are implemented in the wrong order, DDR3 operation will be unpredictable.

Detailed explanations of the registers and their functionality are contained in the [KeyStone Architecture DDR3 Memory Controller User Guide \(SPRUGV8\)](#). Board layout guidance is provided in the [DDR3 Design Requirements for KeyStone Devices Application Report \(SPRABI1\)](#).

All DDR3 initialization routines must contain the basic register writes to configure the memory controller within the DSP as well as register writes that configure the mode registers within the attached DRAM devices. The datasheet for the DRAM implemented must be referenced to optimize these values. Additionally, since DDR3 implementations use a fly-by routing topology, PCB track lengths for the fly-by signals (Address, Command, Control and Clock) and data group signals (DQ, DQS and DQM) must be available to properly initialize the leveling registers.

The spreadsheet discussed in this application report can be downloaded from the following URL: <http://www.ti.com/lit/zip/sprabl2.zip>

## Contents

1	New DDR3 Feature: Leveling	3
1.1	Write and Read Leveling	3
1.2	Automatic Leveling Initialization	3
1.3	Invert Clock Out	3
2	DDR3 Controller Configuration	4
2.1	Header Section	4
2.2	KICK Unlock and DDR3 PLL Configuration	5
2.3	Leveling register configuration	5
2.4	Basic Controller and DRAM Configuration	7
3	Leveling Execution	12
3.1	Fixed Leveling	12
3.2	Partial Automatic Leveling	13
3.2.1	Executing Partial Automatic Leveling	13
3.2.2	Enabling Incremental Leveling with Partial Automatic Leveling	14
3.3	Full Automatic Leveling	14
4	Lock KICK Registers	16
5	Revision History	17



Please be aware that an important notice concerning availability, standard warranty, and use in critical applications of Texas Instruments semiconductor products and disclaimers thereto appears at the end of this document.

## 1 New DDR3 Feature: Leveling

This section provides an introduction to the new DDR3 physical interface concept called leveling. A basic understanding of this technology will help programmers understand the required configuration steps.

The physical DDR3 interface on the KeyStone device is often called the DDR3 PHY. It includes the I/O buffers and all of the logic required to support the DDR3 interface technology. The DDR3 interface circuitry also includes registers and control logic to support the physical DDR3 interface as well as control for the DRAM devices. The terms PHY and Controller are used interchangeably in this document to refer to this circuitry.

The KeyStone DDR3 controller supports three modes of leveling.

- Write leveling
- Read eye training
- Read gate training.

These three specific leveling modes are also generally referred to as write and read leveling. This is discussed more in the next subsection. The following subsection explains that the leveling circuitry in the DDR3 controller must be properly configured for robust operation. The final subsection in this introduction describes the purpose of using an inverted DDR3 clock for certain memory implementations.

### 1.1 Write and Read Leveling

Write and read leveling are new controller features in the JEDEC DDR3 implementation. DDR3 operating frequencies are achieved by allowing the address, control, command and clock nets to be routed in a fly-by arrangement. This allows for optimum signal integrity. However, this results in a different delay from the controller to each DRAM. The write leveling circuitry adaptively estimates this delay and offsets the data group signals so that they arrive at the DRAM coincident with the fly-by signals. This adaptation occurs independently for each byte lane resulting in a unique and optimized offset for each byte lane. The different delays for the clock to each DRAM also causes different launch times for the data group signals during reads. This causes the need for read-eye and read-gate training (leveling).

Write leveling must be executed prior to DRAM writes so that the data and control signals arrive at the DRAM with valid timing relationships. Similarly, read-eye and read-gate training must be completed prior to DRAM reads so that the data will be received and sampled by the KeyStone device at the appropriate time.

### 1.2 Automatic Leveling Initialization

The leveling circuitry within the KeyStone DDR3 controller converges all nine byte lanes simultaneously. This is possible because an initial value for each leveling mode is written into the DDR3 PHY byte lanes prior to convergence. There are two primary initialization values for each byte lane. The first is the WRLVL\_INIT\_RATIO (write leveling initialization ratio) value which is based on the difference between the routed clock length and the routed data strobe length. The second is the GATELVL\_INIT\_RATIO (gate leveling initialization ratio) value which is approximately the round trip delay. This is equal to the routed clock length plus the routed data strobe length plus the read sampling delay. The routed clock length

represents the routed length for all of the fly-by signals and the routed data strobe length represents the routed length for the data group signals. The fly-by signals are the address, command, control and clock signals and the data group signals are the data, data strobe, and data mask signals.

The DDR3 PHY Calc spreadsheet is provided to help users compute the initial values and to translate them into the proper units. The inputs are the routed clock and data strobe lengths. The result values are the initial values in units of DLL taps of which there are 256 per clock period. Additionally, since the initial leveling algorithm only adapts in the positive direction, the initial values are offset 128 DLL steps in the negative direction. The spreadsheet can be downloaded from the following URL: <http://www.ti.com/lit/zip/sprabl2.zip>.

### 1.3 Invert Clock Out

Some DDR3 board layouts have very short fly-by routes to the first DRAM and some have long routes to the first DRAM that form a loop. UDIMM modules always have a long loop. Implementations of individual DRAMs often have a fly-by routing delay to the first DRAM that is about equal to the data group delay to that DRAM. Due to the timing uncertainties in the DDR3 PHY circuitry, whenever the board routing delay for the clock is not longer than the data routing by at least one-quarter of a clock period, leveling may fail. To solve this problem, the DDR3 clock output can be inverted to add an apparent extension of one-half the clock period to the clock net and the other associated fly-by routes. This feature is enabled by setting the INVERT\_CLKOUT bit in one of the PHY control registers.

## 2 DDR3 Controller Configuration

The initialization routines normally have five basic pieces. This is true whether the routine is implemented within a Code Composer Studio (CCS) GEL file or embedded software such as C-code. The five sections are:

- Header section - contains '# define' macros that represent cryptic addresses with useful names,
- KICK unlock and DDR3 PLL configuration - this may reside elsewhere but it must be completed prior to initializing the DDR3 controller and DRAM,
- Leveling register configuration - register writes to configure the write and read leveling circuitry,
- Basic controller and DRAM configuration - register writes that configure the critical timing parameters and mode register parameters similar to those used for DDR2 and
- Leveling execution - register writes to invoke the write and read leveling processes.

### 2.1 Header Section

The header section contains '# define' macros that represent cryptic addresses with useful names. The register addresses can be obtained from the [KeyStone Architecture DDR3 Memory Controller User Guide](#) (SPRUGV8) or the Data Manual for the KeyStone device. [Example 1](#) below shows the addresses as defined for the C6678 DSP. See the Data Manual for your specific device to verify the address map. Note that some of the registers are in the address region starting at 0x2100\_0000 defined for the DDR3 memory controller and that many of the registers are located in the chip-level registers address region starting at 0x0262\_0000. Chip-level registers are also referred to as boot configuration registers in other KeyStone documentation. [Example 1 “Sample Header”](#) below contains all of the register locations.

**Example 1 Sample Header**

```
-----
#define DDR3_BASE_ADDR      (0x21000000)
#define DDR_SDCFG           (* (unsigned int*) (DDR3_BASE_ADDR + 0x00000008))
#define DDR_SDRFC           (* (unsigned int*) (DDR3_BASE_ADDR + 0x00000010))
#define DDR_SDTIM1          (* (unsigned int*) (DDR3_BASE_ADDR + 0x00000018))
#define DDR_SDTIM2          (* (unsigned int*) (DDR3_BASE_ADDR + 0x00000020))
#define DDR_SDTIM3          (* (unsigned int*) (DDR3_BASE_ADDR + 0x00000028))
#define DDR_PMCTL           (* (unsigned int*) (DDR3_BASE_ADDR + 0x00000038))
#define RDWR_LVL_RMP_WIN    (* (unsigned int*) (DDR3_BASE_ADDR + 0x000000D4))
#define RDWR_LVL_CTRL       (* (unsigned int*) (DDR3_BASE_ADDR + 0x000000DC))
#define DDR_ZQCFG           (* (unsigned int*) (DDR3_BASE_ADDR + 0x000000C8))
#define DDR_PHYCTRL         (* (unsigned int*) (DDR3_BASE_ADDR + 0x000000E4))

#define DDR3PLLCTL0         (* (unsigned int*) (0x02620330))
#define DDR3PLLCTL1         (* (unsigned int*) (0x02620334))

#define DATA0_WRLVL_INIT_RATIO (* (unsigned int*) (0x0262040C))
#define DATA1_WRLVL_INIT_RATIO (* (unsigned int*) (0x02620410))
#define DATA2_WRLVL_INIT_RATIO (* (unsigned int*) (0x02620414))
#define DATA3_WRLVL_INIT_RATIO (* (unsigned int*) (0x02620418))
#define DATA4_WRLVL_INIT_RATIO (* (unsigned int*) (0x0262041C))
#define DATA5_WRLVL_INIT_RATIO (* (unsigned int*) (0x02620420))
#define DATA6_WRLVL_INIT_RATIO (* (unsigned int*) (0x02620424))
#define DATA7_WRLVL_INIT_RATIO (* (unsigned int*) (0x02620428))
#define DATA8_WRLVL_INIT_RATIO (* (unsigned int*) (0x0262042C))

#define DATA0_GTLVL_INIT_RATIO (* (unsigned int*) (0x0262043C))
#define DATA1_GTLVL_INIT_RATIO (* (unsigned int*) (0x02620440))
#define DATA2_GTLVL_INIT_RATIO (* (unsigned int*) (0x02620444))
#define DATA3_GTLVL_INIT_RATIO (* (unsigned int*) (0x02620448))
#define DATA4_GTLVL_INIT_RATIO (* (unsigned int*) (0x0262044C))
#define DATA5_GTLVL_INIT_RATIO (* (unsigned int*) (0x02620450))
#define DATA6_GTLVL_INIT_RATIO (* (unsigned int*) (0x02620454))
#define DATA7_GTLVL_INIT_RATIO (* (unsigned int*) (0x02620458))
#define DATA8_GTLVL_INIT_RATIO (* (unsigned int*) (0x0262045C))
```

```
#define DDR3_CONFIG_REG_0      (*(unsigned int*) (0x02620404))
#define DDR3_CONFIG_REG_12     (*(unsigned int*) (0x02620434))
#define DDR3_CONFIG_REG_23     (*(unsigned int*) (0x02620460))
#define DDR3_CONFIG_REG_24     (*(unsigned int*) (0x02620464))

#define KICK0                   (*(unsigned int*) (0x2620038))
#define KICK1                   (*(unsigned int*) (0x262003C))

#define KICK0_UNLOCK           0x83E70B13
#define KICK1_UNLOCK           0x95A4F1E0

#define KICK0_LOCK              0
#define KICK1_LOCK              0
```

**End of Example 1**

## 2.2 KICK Unlock and DDR3 PLL Configuration

The KICK unlock and DDR3 PLL configuration may reside elsewhere in the initialization code but they must be completed prior to initializing the DDR3 controller and DRAM. Prior to writing to the DDR3 Controller configuration registers, the KICK registers need to be unlocked. These may then be locked again after configuration is complete. See the [C66x DSP Bootloader User Guide \(SPRUGY5\)](#) or the device data manual for more information. Software examples provided within the multicore software development kit (MCSDK) have a separate subroutine for DDR3 PLL configuration. This is the recommended implementation solution.

Early versions of the KeyStone devices contain PLL initialization errata that must be resolved through software during the PLL configuration. Information about PLL configuration can be found in the [Phase Locked Loop \(PLL\) Controller for KeyStone Devices User Guide \(SPRUGV2\)](#) and the respective KeyStone Data Manual. Information about the PLL errata can be found in the silicon errata document for the specific KeyStone DSP.

The DDR3 PLL within the KeyStone device receives the reference clock provided at the DDR3CLK input and multiplies it up to the rate desired for the DDR3 interface. Note that the clock rate to the DDR3 memory is half of the data rate (i.e. DDR3-1333 operates with a 666.67 MHz clock).

Example 2 “[Programming DDR3 PLL](#)” will unlock the KICK registers and program the DDR3 PLL to generate a 666.67MHz clock (for DDR3-1333 operation) from an input clock of 66.667 MHz. See the [Phase Locked Loop \(PLL\) Controller for KeyStone Devices User Guide \(SPRUGV2\)](#) for a detailed explanation of the sequence of steps and the necessary delays. Note that the DDR3 clock rate affects timing parameters throughout this application note.

### Example 2 Programming DDR3 PLL

```
DDR3PLLCTL1 |= 0x00000040; //Set ENSAT bit = 1
DDR3PLLCTL0 |= 0x00800000; //Set BYPASS bit = 1
DDR3PLLCTL1 |= 0x00002000; //Set RESET bit = 1
DDR3PLLCTL0 = 0x098804C0; //Configure CLKR, CLKF, CLKOD, BWADJ
DDR3PLLCTL1 &= ~(0x0000000F); // Clear upper BWADJ
for(i=0;i<7000;i++); //Wait >5us for reset to complete
DDR3PLLCTL1 &= ~(0x00002000); //Clear RESET bit
for(i=0;i<70000;i++); //Wait >2000 (>50us) clocks for PLL lock
DDR3PLLCTL0 &= ~(0x00800000); //Clear BYPASS bit = 0
```

**End of Example 2**

## 2.3 Leveling Register Configuration

The chip-level registers that provide initialization values to the leveling circuitry need to be programmed next. These values will be used after the basic controller and DRAM configuration but they must be written before that part of the initialization sequence. Values need to be written to the registers DDR3\_CONFIG\_REG\_0, DDR3\_CONFIG\_REG\_12 and possibly DDR3\_CONFIG\_REG\_23 to properly configure the leveling processes. Additionally, initial values need to be written to the WRLVL\_INIT\_RATIO and GATLVL\_INIT\_RATIO registers for all of the data lanes in use when automatic leveling is used.

The INVERT\_CLKOUT bit discussed previously is one of these configuration values. This feature is enabled by setting the INVERT\_CLKOUT bit in the DDR3\_CONFIG\_REG\_12 register. Note that when we set INVERT\_CLKOUT to 1, the CTRL\_SLAVE\_RATIO field in DDR3\_CONFIG\_REG\_0 should be programmed to 0x100. If the INVERT\_CLKOUT bit remains at 0, the default value of the CTRL\_SLAVE\_RATIO field must be programmed to 0x80. [Example 3](#) sets the INVERT\_CLKOUT bit and programs the CTRL\_SLAVE\_RATIO field properly.

### Example 3 INVERT\_CLKOUT Programming

```
-----
DDR3_CONFIG_REG_0  &= ~(0x007FE000); // clear ctrl_slave_ratio field
DDR3_CONFIG_REG_0  |= 0x00200000;    // set ctrl_slave_ratio to 0x100
DDR3_CONFIG_REG_12 |= 0x08000000;    // Set invert_clkout = 1
-----
```

End of Example 3

The DLL\_LOCK\_DIFF field is also programmed at this time to 0xF since it is the only other programmable value in the DDR3\_CONFIG\_REG\_0 register. The DLL\_LOCK\_DIFF field must always be written with this value.

### Example 4 DLL\_LOCK\_DIFF Programming

```
-----
DDR3_CONFIG_REG_0 |= 0xF; // set dll_lock_diff to 15
-----
```

End of Example 4

Additional writes to DDR3\_CONFIG\_REG\_23 and DDR3\_CONFIG\_REG\_24 would be inserted here depending on the leveling mode chosen. This topic is explained in [Section 3](#) of this application note

The next group of chip-level registers that needs to be written are the initial values for the automatic leveling process (when it is used). The DDR3 PHY Calc spreadsheet is provided to help compute the initial values and to translate them into the proper units. The inputs are the routed clock and data strobe lengths. The result values are the initial values in units of DLL taps. Follow the instructions in the DDR3 PHY Calc spreadsheet to generate the set of initial values for the board implementation. The initial values below are those computed from the C6678 EVM design.

### Example 5 Leveling Initialization Values

```
-----
DATA0_WRLVL_INIT_RATIO = 0x5E;
DATA1_WRLVL_INIT_RATIO = 0x5E;
DATA2_WRLVL_INIT_RATIO = 0x5E;
DATA3_WRLVL_INIT_RATIO = 0x51;
DATA4_WRLVL_INIT_RATIO = 0x38;
DATA5_WRLVL_INIT_RATIO = 0x3A;
DATA6_WRLVL_INIT_RATIO = 0x24;
DATA7_WRLVL_INIT_RATIO = 0x20;
DATA8_WRLVL_INIT_RATIO = 0x44;

DATA0_GTLVL_INIT_RATIO = 0xDD;
DATA1_GTLVL_INIT_RATIO = 0xDD;
-----
```

```
DATA2_GTLVL_INIT_RATIO = 0xBE;
DATA3_GTLVL_INIT_RATIO = 0xCA;
DATA4_GTLVL_INIT_RATIO = 0xA9;
DATA5_GTLVL_INIT_RATIO = 0xA7;
DATA6_GTLVL_INIT_RATIO = 0x9E;
DATA7_GTLVL_INIT_RATIO = 0xA1;
DATA8_GTLVL_INIT_RATIO = 0xBA;
```

**End of Example 5**

Lastly, the PHY\_RESET is pulsed (0 -> 1 -> 0) to latch these leveling configuration values into the PHY logic.

#### Example 6 Pulsing the PHY\_RESET

```
DDR_PHYCTRL &= ~(0x00008000);
DDR_PHYCTRL |= (0x00008000);
DDR_PHYCTRL &= ~(0x00008000);
```

**End of Example 6**

## 2.4 Basic Controller and DRAM Configuration

The basic controller and DRAM configuration writes occur next. These register writes configure the controller critical timing parameters and the DRAM mode register parameters. The [KeyStone Architecture DDR3 Memory Controller User Guide \(SPRUGV8\)](#) must be used to create these configuration values. The datasheet timing from the DRAM will also need to be referenced to extract critical timing parameters. Formulas and look-up tables (LUTs) in the DDR3 Controller User's Guide provide the translation from datasheet parameters to the bit-field values. A DDR3 Register Calc spreadsheet is also available to streamline this process. It can be downloaded from the following URL: <http://www.ti.com/lit/zip/sprabl2.zip>.

The programmed SDRAM timing values below are based on implementation of a 64-bit memory rank composed of four 16-bit SDRAMs. The parts used in this example are 1Gb DRAMs from Samsung (K4B1G1646E-HCH9). The DDR3 clock rate is 666MHz. These are the DRAMs and the topology implemented on the C6678 EVM. [Table 1](#) shows the relevant timing parameters extracted from the DRAM datasheet along with their calculated bit field values.

**Table 1 SDRAM Timing Register Values**

Controller Register	Datasheet Timing Parameter	Datasheet Value	Calculated Value (Hexadecimal)
<b>SDTIM1</b>	T_RP	13.5ns	0x8
	T_RCD	13.5ns	0x8
	T_WR	15ns	0x9
	T_RAS(min)	36ns	0x17
	T_RC	49.5ns	0x20
	T_RRD (use T_FAW since 8 banks)	45ns	0x7
	T_WTR	7.5ns	0x4
<b>SDTIM2</b>	T_RTP	7.5ns	0x4
	T_CKE	5.625ns	0x3
	T_XP	6ns	0x3
	T_XS T_XSNR	120tCK	0x4F
	T_XSDLL T_XSRD	512tCK	0x1FF
<b>SDTIM3</b>	T_RFC (min)	110ns	0x49
	T_ZQCS	64tCK	0x3F
	T_CKESR	5.625ns	0x4



There are several other parameters that must be configured which reside in the SDCFG register. These include impedance settings which must match the board and SDRAM design and SDRAM size parameters. Some of these affect the operation of the DDR3 memory interface of the DSP and some of them affect the values programmed into the SDRAM Mode Registers. The [KeyStone Architecture DDR3 Memory Controller User Guide](#) (SPRUGV8) must be referenced to determine the proper bit field values for this register from the LUTs provided. [Table 2](#) provides an overview of these parameters and shows the choices implemented in the C6678 EVM.

**Table 2 SDRAM Configuration Register Values**

Controller Register	Interface Parameter Field	Option Chosen	Value from LUT (Hexadecimal)
SDCFG	IBANK_POS	8 banks	0x0
	DDR_TERM	RZQ/6 $\Omega$	0x3
	DYN_ODT	Disable	0x0
	SDRAM_DRIVE	RZQ/7 $\Omega$	0x1
	CWL	CWL = 7 clocks	0x2
	NM	64-bit bus width	0x0
	CL	CAS = 9 clocks	0xA
	ROWSIZE	13 row bits	0x4
	IBANK	8 banks	0x3
	EBANK	DCE0#	0x0
	PAGESIZE	1024-word page	0x2

Each of the register writes will be discussed individually in the recommended order.

The DDR\_SDRFC register controls the behavior of refresh. To achieve proper initialization timing, the refresh interval must be 31.25us rather than the normal 7.8us interval to create the necessary initial CKE low period of 500us. The 0x5162 portion of this register sets the refresh period to 31.25us with the current clock rate. The MSB of this register may be set or cleared at this time to enable or disable SDRAM configuration. It will be cleared at a later time to enable configuration when desired.

**Example 7 Refresh Control Register Programming #1**

```
-----
DDR_SDRFC = 0x00005162; // enable configuration
```

**End of Example 7**

The DDR\_SDTIM[3:1] registers contain most of the critical DRAM and controller timing parameters as shown in [Table 1](#). See the [DDR3 Memory Controller for KeyStone Devices User Guide](#) (SPRUGV8) and the SDRAM device datasheet to determine the values to write into these registers. These values are based on the timing parameters extracted from the DRAM datasheet. The sequences in [Example 8](#) compute and assign the DDR\_SDTIM[3:1] values.

**Example 8 SDRAM Timing Register Computation and Programming**

```
-----
TEMP = 0;
TEMP = 0x8 << 25; // T_RP bit field 28:25
TEMP = 0x8 << 21; // T_RCD bit field 24:21
TEMP = 0x9 << 17; // T_WR bit field 20:17
TEMP = 0x17 << 12; // T_RAS bit field 16:12
TEMP = 0x20 << 6; // T_RC bit field 11:6
TEMP = 0x7 << 3; // T_RRD bit field 5:3
TEMP = 0x4; // T_WTR bit field 2:0
DDR_SDTIM1 = TEMP;
```



```

TEMP = 0;
TEMP |= 0x3 << 28; // T_XP bit field 30:28
TEMP |= 0x4f << 16; // T_XSNR bit field 24:16
TEMP |= 0x1ff << 6; // T_XSRD bit field 15:6
TEMP |= 0x4 << 3; // T_RTP bit field 5:3
TEMP |= 0x3; // T_CKE bit field 2:0
DDR_SDIM2 = TEMP;

TEMP = 0;
TEMP |= 0x5 << 28; // T_PDLL_UL bit field 31:28 (fixed value)
TEMP |= 0x5 << 24; // T_CSTA bit field 27:24 (fixed value)
TEMP |= 0x4 << 21; // T_CKESR bit field 23:21
TEMP |= 0x3f << 15; // T_ZQCS bit field 20:15
TEMP |= 0x49 << 4; // T_RFC bit field 12:4
TEMP |= 0xf; // T_RAS_MAX bit field 3:0 (fixed value)
DDR_SDIM3 = TEMP;

```

#### End of Example 8

Alternately, the DDR\_SDIM[3:1] values can be computed manually from guidance in the [KeyStone Architecture DDR3 Memory Controller User Guide](#) (SPRUGV8) or the DDR3 Register Calc spreadsheet and written with the computed values. The equations above will yield the values shown below. However, the equations above are more useful if any of the individual fields are expected to change while tuning the board configuration.

#### Example 9 SDRAM Timing Register Programming

```

DDR_SDIM1 = 0x1113783C;
DDR_SDIM2 = 0x304F7FE3;
DDR_SDIM3 = 0x559F849F;

```

#### End of Example 9

DDR\_PHYCTRL, also known as DDR\_PHY\_CTRL\_1 or DDR\_PHYC, programs termination modes for the output buffers and the read latency in the lower five bits. The values supported in the upper bits are given and cannot be changed on customer designs. The read latency is somewhat based on board layout. A conservative value for it is CL+3 since the controller can operate with up to 4 clocks of round trip delay. The value of 0xF chosen below adds extra margin but also impacts read-to-write turn-around performance. The value 0xC can be used in this field since we have previously programmed CL=9.

#### Example 10 PHY Control Register Programming

```

DDR_PHYCTRL = 0x0010010F;

```

#### End of Example 10

The DDR\_ZQCFG register configures the adaptive impedance calibration capability. This is a new capability in DDR3. Based on a 240 ohm resistor attached to each DRAM, the DRAM can re-calibrate its output impedance periodically as its temperature changes. This example causes the ZQ calibration to occur once every 100ms. (This interval is under study.) See the *DDR3 Controller User's Guide* for more details.

#### Example 11 Dynamic Impedance Configuration Register Programming

```

DDR_ZQCFG = 0x70073214;

```

#### End of Example 11

PMCTL is the power management control register. Writing a 0 to it disables power management. Refer to the *DDR3 Controller User Guide* for more information on the supported features. Programming for the power management control register, latency configuration register, class of service registers, interrupt enable registers and the ECC control registers can be done here or at a later time if these features are needed. Programming these registers is beyond the scope of this application note.

#### Example 12 Power Management Register Programming

```
-----
DDR_PMCTL    = 0x0;
```

#### End of Example 12

The additional write to DDR\_SDRFC has the MSB cleared which enables DRAM configuration. It still has the refresh interval programmed to the longer number needed during DRAM initialization.

#### Example 13 Refresh Control Register Programming #2

```
-----
DDR_SDRFC    = 0x00005162;    // enable configuration
```

#### End of Example 13

A write to DDR\_SDCFG completes the controller configuration and also causes the mode registers in the DRAM memories to be programmed. The hardware configuration of the DRAM devices occurs immediately since the MSB in DDR\_SDRFC is cleared.

The DDR\_SDCFG register contains other DRAM and controller configuration parameters as shown in [Table 2](#). See the *DDR3 Memory Controller for KeyStone Devices User Guide* (SPRUGV8) and the SDRAM device datasheet to determine the values to write into these registers. [Example 14](#) computes and assigns the DDR\_SDCFG value.

#### Example 14 SDRAM Configuration Register Computation and Programming

```
-----
TEMP = 0;
TEMP = 0x3 << 29;    // SDRAM_TYPE bit field 31:29 (fixed value)
TEMP = 0x0 << 27;    // IBANK_POS bit field 28:27
TEMP = 0x3 << 24;    // DDR_TERM bit field 26:24
TEMP = 0x0 << 21;    // DYN_ODT bit field 22:21
TEMP = 0x1 << 18;    // SDRAM_DRIVE bit field 19:18
TEMP = 0x2 << 16;    // CWL bit field 17:16
TEMP = 0x0 << 14;    // NM bit field 15:14
TEMP = 0xA << 10;    // CL bit field 13:10
TEMP = 0x4 << 7;     // ROWSIZE bit field 9:7
TEMP = 0x3 << 4;     // IBANK bit field 6:4
TEMP = 0x0 << 3;     // EBANK bit field 3:3
TEMP = 0x2;          // PAGESIZE bit field 2:0
DDR_SDCFG = TEMP;
```

#### End of Example 14

Alternately, the DDR\_SDCFG value can be computed manually from guidance in the *DDR3 Memory Controller for KeyStone Devices User Guide* (SPRUGV8) or the DDR3 Register Calc spreadsheet and written with the computed values. The equations in [Example 14](#) will yield the value shown below in [Example 15](#). However, the equations above are more useful if any of the individual fields are expected to change while tuning the board configuration.

#### Example 15 SDRAM Configuration Register Programming

```
-----
DDR_SDCFG    = 0x63062A32;    // last config write - DRAM init occurs
```

#### End of Example 15

A delay loop will be needed at this point to force the GEL/software initialization sequence to pause while the hardware initialization completes. The 600us minimum delay is required to allow the SDRAM devices to be completely initialized. After hardware initialization completes, the refresh interval can be reprogrammed to the standard operating rate.

**Example 16 Refresh Control Register Programming #3**  
-----

```
for(i=0;i<1000;i++);          //Wait 600us for HW init to complete
DDR_SDRFC = 0x00001450;        //Refresh rate = (7.8*666MHz]
```

**End of Example 16**  
-----

### 3 Leveling Execution

Leveling execution is triggered by the last group of register writes. These writes initialize the write and read leveling circuitry to prepare the DDR byte lanes in the PHY for data transfer.

There are three different DDR PHY write and read leveling operating modes supported in the KeyStone DSPs.

- Fixed leveling - a single set of delay values are used for all nine byte lanes (also known as 'ratio forced' leveling)
- Partial automatic leveling - automatic write leveling and read gate training with a fixed read eye sample point that may be used with incremental leveling for the read gate
- Full automatic leveling - automatic write leveling and read gate training with an adaptive read eye sample point that is achieved with read eye incremental leveling

The last two modes can work with incremental read leveling. Incremental leveling allows the circuitry within the DDR PHY to adapt to delay changes in the DSP, PCB traces and DRAM to optimize the read sampling.

The leveling circuitry measures the delays in the read and write paths. These delays are different from device to device and they also vary when the temperature and voltage change. The device to device variation creates the need for the automatic leveling at system initialization. The read delay variations due to temperature and voltage changes may be compensated for by incremental leveling. Incremental leveling can occur periodically so that the leveling parameters track during full memory operation. This allows the memory interface to operate at higher speeds for longer periods of time without fault. Incremental leveling is highly recommended for long-term DDR3 operation stability at the higher operating rates.

Initial silicon errata limit the leveling modes available on some devices. [Table 3](#) shows the leveling modes available in the initial versions of the KeyStone DSPs.

**Table 3**      **Leveling Modes Supported**

KeyStone Device	Rev.	Fixed Leveling	Partial Automatic Leveling	Partial Automatic & Incremental Leveling	Full Automatic & Incremental Leveling
TCI6616	PG1.0	X			
TCI6602, TCI6604, TCI6608, C6671, C6672, C6674 and C6678	PG1.0	X	X	X	
TCI6618 and C6670	PG1.0 & PG2.0	X	X	X	X
TCI6602, TCI6604, TCI6608, C6671, C6672, C6674 and C6678	PG2.0				
TCI6612 and TCI6614	PG2.0				
C6655 and C6657	PG1.0				

#### 3.1 Fixed Leveling

The original DDR3 PHY design had a weakness that allowed the read data eye leveling to fail to converge under some conditions. Since all of the leveling modes were combined, that prevented automatic leveling from being functional. Fixed leveling is the only solution for DDR3 operation on PG1.0 TCI6616 DSPs.

Fixed leveling (also known as ratio forced leveling) is only a partial solution. In this mode, the DRAM interface operates with automatic leveling circuitry disabled. KeyStone DSPs only have a single set of fixed leveling registers for all nine byte lanes. Therefore, fixed leveling must be used with a mid-point value for each of the leveling parameters and that will be sub-optimal for most byte lanes. The error between these values and the corresponding delays at the first and last DRAM memories consumes part of the data sample window or 'eye'. This results in a reduced speed of operation. For this reason, we recommend limiting fixed leveling to DDR3-800 with a 400MHz DDR3 clock.

The values used are the midpoints of the estimated values from the DDR3 PHY Calc spreadsheet. These values are then written into the DDR3\_CONFIG\_REG\_23 and DDR3\_CONFIG\_REG\_24 registers. [Example 17](#) completes this programming. These lines would be placed at the end of the leveling register configuration discussed in [Section 2.3](#), and the byte lane initialization values would not be needed.

#### Example 17 Fixed Level Programming

```
-----
#define RD_QOS_SLAVE_RATIO 0x34
#define WR_QOS_SLAVE_RATIO 0xBF
#define WR_DATA_SLAVE_RATIO 0xFF
#define FIFO_WE_SLAVE_RATIO 0x13D

DDR3_CONFIG_REG_23 = RD_QOS_SLAVE_RATIO;
DDR3_CONFIG_REG_23 |= (WR_QOS_SLAVE_RATIO << 10);
DDR3_CONFIG_REG_23 |= (WR_DATA_SLAVE_RATIO << 20);
DDR3_CONFIG_REG_24 = FIFO_WE_SLAVE_RATIO;
```

End of Example 17



**Note**—Incremental leveling is not supported in this mode.

## 3.2 Partial Automatic Leveling

An enhancement was added to the previous DDR3 PHY implementation to fix the read eye sample point problem. This implementation allows the read data eye sample point to be programmed to a fixed value while the other portions of the leveling circuitry could converge as originally intended. This is the hybrid solution referred to as partial automatic leveling.

Since the write leveling and the read gate training can now converge automatically independent of the read eye sample point, the read gate timing can also support incremental adjustments to compensate for timing changes due to voltage and temperature changes. Incremental leveling for the write leveling and read eye sample point are not supported in this version of the DDR3 PHY.

### 3.2.1 Executing Partial Automatic Leveling

Partial automatic leveling needs all of the byte lane initialization values programmed in “[2.3 Leveling Register Configuration](#)” on page 6 prior to the controller and DRAM initialization. It also needs the following write to DDR3\_CONFIG\_REG\_23 during the leveling register configuration steps.

#### Example 18 Partial Automatic Leveling Programming

```
-----
DDR3_CONFIG_REG_23 |= 0x00000200;
```

End of Example 18

After the controller and DRAM initialization is complete, automatic leveling can be triggered. The remaining register writes simply enable and then start the initialization process. The RDWR\_LVL\_EN field is the MSB of RDWR\_LVL\_RMP\_CTRL and the RDWR\_LVL\_FULL\_START field is the MSB of RDWR\_LVL\_CTRL. Note that the other bit fields in these registers are all cleared to zero. These can be programmed at a later time to enable incremental leveling.

The writes in [Example 19](#) trigger automatic convergence of the write leveling and the read gate training while leaving the default fixed value for the read eye sample point as indicated by the prior write to DDR3\_CONFIG\_REG\_23. Note that the 3ms stall after enabling leveling is required for proper operation.

#### Example 19 Partial Automatic Leveling Enable

```
-----
RDWR_LVL_RMP_CTRL = 0x80000000;
RDWR_LVL_CTRL = 0x80000000;
for(i=0; i<1000; i++); //Wait 3ms for leveling to complete
```

End of Example 19

If there is concern that the leveling is not converging correctly, the timeout register bits can be read at this time to verify that the leveling completed as expected. The leveling timeout indications are bits 4, 5 and 6 in the DDR3 memory controller status register at address 0x21000004. If any of these bits are set to a one, leveling has failed. This normally means there is either a hardware problem or the initial leveling values are incorrect. There is also an IFRDY bit in this register that will be set if leveling completes successfully.

### 3.2.2 Enabling Incremental Leveling with Partial Automatic Leveling

Incremental leveling after partial automatic leveling is optional. Separate control is available to enable or disable incremental adjustments to each of the leveling modes: write leveling, read eye training and read gate training. Incremental leveling after partial automatic leveling can only be enabled for read gate training. See the [DDR3 Memory Controller for KeyStone Devices User Guide](#) (SPRUGV8) for a detailed explanation of these fields.

The following writes in [Example 20](#) will enable incremental gate training. It will perform these incremental adjustments on a 30us interval during a Smart Reflex voltage change and a 10ms interval during normal operation. The ramp window is also set to 10ms. (These intervals are under study.)

#### Example 20 Incremental Leveling After Partial Automatic Leveling

```
-----
RDWR_LVL_RMP_WIN = 0x00000502;
RDWR_LVL_RMP_CTRL = 0x80000300;
RDWR_LVL_CTRL = 0xFF000900;
```

End of Example 20

## 3.3 Full Automatic Leveling

The latest KeyStone devices contain additional enhancement to allow all three leveling modes to be used. The write leveling, read data eye training and read gate training are all triggered for initial convergence. However, the read eye sample point may still be invalid. Incremental leveling will then force the read eye sample point to a good starting value and then it will robustly optimize the read eye sample point after multiple successive iterations. Simultaneously, read gate timing will also robustly track voltage and temperature driven timing changes during the incremental leveling iterations.

Full automatic leveling needs all of the byte lane initialization values programmed in the leveling register configuration section prior to the controller and DRAM initialization. There must be no writes to DDR3\_CONFIG\_REG\_23 when full automatic leveling is used.

Once the controller and DRAM initialization is complete, the remaining register writes simply enable and then start the initialization process. The RDWR\_LVL\_EN field is the MSB of RDWR\_LVL\_RMP\_CTRL and the RDWR\_LVL\_FULL\_START field is the MSB of RDWR\_LVL\_CTRL. Note that the other bit fields in these registers are all cleared to zero. These will be programmed at a later time to enable incremental leveling. Note that the 3ms stall after enabling leveling is required for proper operation.

#### **Example 21 Full Automatic Leveling Enable**

```
RDWR_LVL_RMP_CTRL = 0x80000000;
RDWR_LVL_CTRL = 0x80000000;
for(i=0; i<1000; i++); //Wait 3ms for leveling to complete
```

#### **End of Example 21**

If there is concern that the leveling is not converging, the timeout register bits can be read at this time to verify that the leveling completed as expected. The leveling timeout indications are bits 4, 5 and 6 in the DDR3 Memory Controller Status Register at address 0x21000004. If any of these bits are set to a one, leveling has failed. This normally means there is either a hardware problem or the initial leveling values are incorrect. There is also an IFRDY bit in this register that will be set if leveling completes successfully.

Incremental leveling of at least the read eye sample point must be executed at least 64 times after full automatic leveling to converge it to an initial optimum value. Separate control is available to enable or disable incremental adjustments to each of the leveling modes - write leveling, read eye training and read gate training. Incremental leveling after a full automatic leveling may be enabled for read eye and read gate training. See the [DDR3 Memory Controller for KeyStone Devices User Guide](#) (SPRUGV8) for a detailed explanation of these fields.

The writes in [Example 22](#) will enable the read eye and read gate training modes of incremental leveling. It will perform these incremental adjustments on a 30us interval during a Smart Reflex voltage change and a 10ms interval during normal operation. The ramp window is also set to 10ms. (These intervals are under study.)

#### **Example 22 Incremental Leveling After Full Automatic Leveling**

```
RDWR_LVL_RMP_WIN = 0x00000502;
RDWR_LVL_RMP_CTRL = 0x80030300;
RDWR_LVL_CTRL = 0xFF090900;
```

#### **End of Example 22**



## 4 Lock KICK Registers

At the end of the DDR3 controller configuration, the KICK registers may be locked. This can be done immediately after the DDR3 controller configuration or later after other memory regions protected by this mechanism are completely configured.

### Example 23 Lock Kick Registers

```
-----
KICK0 = KICK0_LOCK;           //lock kicker registers
KICK1 = KICK1_LOCK;
```

### End of Example 23

Note that the need to lock the KICK registers is application dependent. If the application will allow multiple cores to access the bootcfg registers, there may be race conditions (see the chip-specific Errata document). One workaround for this problem is not locking the KICK registers at the end of this process.

## 5 Revision History

The following table lists changes for each revision.

Revision	Description of Changes
0	Initial Release



## IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

TI products are not authorized for use in safety-critical applications (such as life support) where a failure of the TI product would reasonably be expected to cause severe personal injury or death, unless officers of the parties have executed an agreement specifically governing such use. Buyers represent that they have all necessary expertise in the safety and regulatory ramifications of their applications, and acknowledge and agree that they are solely responsible for all legal, regulatory and safety-related requirements concerning their products and any use of TI products in such safety-critical applications, notwithstanding any applications-related information or support that may be provided by TI. Further, Buyers must fully indemnify TI and its representatives against any damages arising out of the use of TI products in such safety-critical applications.

TI products are neither designed nor intended for use in military/aerospace applications or environments unless the TI products are specifically designated by TI as military-grade or "enhanced plastic." Only products designated by TI as military-grade meet military specifications. Buyers acknowledge and agree that any such use of TI products which TI has not designated as military-grade is solely at the Buyer's risk, and that they are solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI products are neither designed nor intended for use in automotive applications or environments unless the specific TI products are designated by TI as compliant with ISO/TS 16949 requirements. Buyers acknowledge and agree that, if they use any non-designated products in automotive applications, TI will not be responsible for any failure to meet such requirements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

### Products

Audio	<a href="http://www.ti.com/audio">www.ti.com/audio</a>
Amplifiers	<a href="http://amplifier.ti.com">amplifier.ti.com</a>
Data Converters	<a href="http://dataconverter.ti.com">dataconverter.ti.com</a>
DLP® Products	<a href="http://www.dlp.com">www.dlp.com</a>
DSP	<a href="http://dsp.ti.com">dsp.ti.com</a>
Clocks and Timers	<a href="http://www.ti.com/clocks">www.ti.com/clocks</a>
Interface	<a href="http://interface.ti.com">interface.ti.com</a>
Logic	<a href="http://logic.ti.com">logic.ti.com</a>
Power Mgmt	<a href="http://power.ti.com">power.ti.com</a>
Microcontrollers	<a href="http://microcontroller.ti.com">microcontroller.ti.com</a>
RFID	<a href="http://www.ti-rfid.com">www.ti-rfid.com</a>
OMAP Mobile Processors	<a href="http://www.ti.com/omap">www.ti.com/omap</a>
Wireless Connectivity	<a href="http://www.ti.com/wirelessconnectivity">www.ti.com/wirelessconnectivity</a>

### Applications

Automotive and Transportation	<a href="http://www.ti.com/automotive">www.ti.com/automotive</a>
Communications and Telecom	<a href="http://www.ti.com/communications">www.ti.com/communications</a>
Computers and Peripherals	<a href="http://www.ti.com/computers">www.ti.com/computers</a>
Consumer Electronics	<a href="http://www.ti.com/consumer-apps">www.ti.com/consumer-apps</a>
Energy and Lighting	<a href="http://www.ti.com/energy">www.ti.com/energy</a>
Industrial	<a href="http://www.ti.com/industrial">www.ti.com/industrial</a>
Medical	<a href="http://www.ti.com/medical">www.ti.com/medical</a>
Security	<a href="http://www.ti.com/security">www.ti.com/security</a>
Space, Avionics and Defense	<a href="http://www.ti.com/space-avionics-defense">www.ti.com/space-avionics-defense</a>
Video and Imaging	<a href="http://www.ti.com/video">www.ti.com/video</a>

TI E2E Community Home Page

[e2e.ti.com](http://e2e.ti.com)

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265  
Copyright © 2012, Texas Instruments Incorporated