NXP Semiconductors
Engineering Bulletin

Document Number: EB828
Rev. 0, 08/2016

MMDC & NoC Configuration for Optimal DDR3 Performance on the i.MX 6DualPlus/6QuadPlus

1 Introduction

This document briefly describes the NXP recommended configuration of the DDR3 interface on the i.MX 6DualPlus and 6QuadPlus SoCs for optimal performance and to better align parameters as specified in the JEDEC DDR3 SDRAM Standard JESD79-3. Configuration of certain registers in the Multi Mode DDR Controller (MMDC), IOMUX, and NoC modules is required in the software initialization of the DDR interface.

The document also briefly describes additional methods to further optimize certain parameters such as SDCLK and DQS duty cycle and VIX parameters, however these are not absolutely necessary to bring them into the JEDEC specification and may have limited applicability based on the customer designs. The LPDDR2 interface programming and optimization are not covered in this document and is to be covered in the future.

Contents

2 MMDC register programming

The i.MX 6DualPlus/6QuadPlus Multi Mode DDR Controller (MMDC) can be used to program the DDR3/DDR3L device for proper operation. This is achieved by an initialization sequence of specific register writes prior to accessing the external DDR device. Programming of the DDR3/DDR3L memory device is dependent on various factors such as:

- DDR memory timing and speed grade
- Memory layout (Fly-by, T topology)
- Bus width (x32, x64)
- Drive Strength and Board layout

Since the above factors are dependent on the customer's DDR memory selection, use case and board design, common programming recommendations cannot be provided as they will be unique for each customer design. NXP provides a DDR3 MMDC register programming aid to help in configuring these specific parameters. Contact your NXP field engineer or sales representative for the i.MX 6DualPlus/6QuadPlus DDR3 Register Programming Aid.

2.1 MMDC programming sequence

The MMDC Controller performs the DRAM initialization sequence according to the JEDEC standard. See the JEDEC spec for the initialization sequence. The recommend steps for programming the MMDC to configure the DDR3/DDR3L memory device for proper operation are listed below:

1. Preliminary: Turn on all necessary clocks.
2. Program IOMUX Pad setting registers.
3. Program Calibrations Setting registers.
   — End calibration issue *frc_msr* to set calibration values in the MMDC PHY.
4. Program MMDC timing parameters in core.
   — Set Configuration Request *(con_req)* bit in MMDC0_MDSCR.
5. Program DDR3/DDR3L memory device mode registers.
6. Program Miscellaneous SoC parameters including NoC registers.
7. Last Step: Allows AXI bus to begin transferring data to MMDC and DDR memory device.
   — End MMDC initialization by clearing the Configuration Request *(con_req)* bit.

NOTE

Using an incorrect programming sequence can cause various memory stability issues; follow the outlined DDR initialization sequence.

There are common parameters required for configuring the DDR interface that are silicon dependent. The programming of these specific parameters for the i.MX 6DualPlus/6QuadPlus are described in the following sections.

3 SDCLK & DQS duty cycle fine tuning

The i.MX 6DualPlus/6QuadPlus DDR Controller (MMDC) has the ability to fine tune the duty cycle of the DRAM_SDCLK0 and DRAM_SDCLK1 clock signals as well as the DQS clock signals.

This is accomplished by modifying the MMDC Duty Cycle Control Register 1 (MMDC1_MPDCCR) for DRAM_SDCLK0 and MMDC Duty Cycle Control Register 2 (MMDC2_MPDCCR) for DRAM_SDCLK1 respectively. MMDC1_MPDCCR register is mapped to AXI channel 0 and MMDC2_MPDCCR is mapped to AXI channel 1.

Programming of these registers after initialization is only permitted by entering the DDR3 device into self-refresh mode through LPMD/DVFS mechanism. Therefore it is recommended to modify these registers in the initial MMDC configuration performed by the software boot loader such as UBOOT.

The following bits in the MMDC Duty Cycle Control Registers are used to control the duty cycle of the clock (DRAM_SDCLK0/1).

- CK_FT0_DCC: Primary duty cycle fine tuning control of the DDR clock
- CK_FT1_DCC: Secondary duty cycle fine tuning control of the DDR clock

Settings of these register bits are as follows:

Table 1. MMDC duty cycle control registers - CK_FTx_DCC bit settings

CK_FTx_DCC Bit Setting	Duty Cycle Impact	Notes
001b	Reduce by 1.5%	Actual duty cycle reduction may vary
010b	No Change	Out of reset default register setting
100b	Increase by 1.5%	Actual duty cycle change may vary

The adjustments are cascaded which means that adjustment FT0 is applied first, and then adjustment FT1 is applied to the result of the first adjustment. If an adjustment of 1.5% is desired, either stage CK_FT0_DCC or stage CK_FT1_DCC can be used. If a larger adjustment is desired, then both stages CK_FT0_DCC and CK_FT1_DCC are used, and applied in the same direction.

Read and write DQS duty cycle adjustments can be achieved by modifying the RD_DQSx_FT_DCC WR_DQSx_FT_DCC bits in the MMDC Duty Cycle Control Register 1 (MMDC1_MPDCCR) for Bytes 0-3 and MMDC Duty Cycle Control Register 2 (MMDC2_MPDCCR) for Bytes 4-7 respectively.

Settings of these DQS fine tune adjustment register bits are as follows:

Table 2. MMDC DQS duty cycle control registers - RD/WR_DQS_FTx_DCC bit settings

RD/WR_DQS_FTx_DCC Bit Setting	Duty Cycle Impact	Notes
001b	Increase by 1.5%	Duty cycle impact of the RD/WR_DQS_FTx_DCC bits are opposite to the SDCLK CK_FTx_DCC Bit Setting register. Actual duty cycle reduction may vary
010b	No Change	Out of reset default register setting
100b	Reduce by 1.5%	Actual duty cycle change may vary

3.1 Recommended SDCLK & DQS duty cycle fine tune settings

Design simulations combined with actual circuit measurements on NXP hardware, over worst case silicon process and temperature conditions were performed to derive the optimal settings for Duty Cycle performance. The following DQS Duty Cycle settings also help to optimize the tDQSH/L parameters and bring them closer to the JEDEC defined limits.

NOTE

The following recommendations are based on measurements on NXP hardware and are presented as guidelines for the customer. Variations due to the different board layouts, topology, memory selection, components and other various application-specific use case factors can yield slightly different results. Customers are recommended to optimize parameters for their end product appropriately.

The settings in the following tables, give the best margin on memory stress testing with Auto ZQ calibration enabled at the maximum supported 528 MHz DDR clock frequency.

Table 3. Recommended duty cycle fine tune parameter settings

Clock	Register	CK_FT0_DCC	CK_FT1_DCC	Notes
SDCLK0	MMDC1_MPDCCR Address: 0x021B_08C0	010b	001b	Since CK_FT0_DCC = 0x2 is the default value out of reset, only modify CK_FT1_DCC to 0x1
SDCLK1	MMDC2_MPDCCR Address 0x021B_48C0	100b	001b	SDCLK1 has a different optimization than SDCLK0. Configuration of SDCLK1 in MMDC2_MPDCCR is not required for applications not using SDCLK1

NOTE

The above recommended is for maximum supported 528 MHz DDR clock frequency. Customers may need to optimize settings for other DDR Clock frequencies used in their respective designs.

Table 4. Recommended DQS duty cycle fine tune parameter settings

Parameter	Register	RD_FTx_DCC	WR_FTx_DCC	Notes
DQS 2-3	MMDC1_MPDCCR Address: 0x021B_08C0	010b	010b	—
DQS 0-1	MMDC1_MPDCCR Address: 0x021B_08C0	010b	001b	Increase by 1.5% for the WR_DQS duty Cycle. No change for RD_DQS
DQS 4,5,7	MMDC2_MPDCCR Address 0x021B_48C0	010b	010b	No modifications are proposed for DQS 4, 5 and 7 Write DQS Duty Cycle tuning (read or write strobes)
DQS 6	MMDC2_MPDCCR Address 0x021B_48C0	010b	001b	Increase by 1.5% for the WR_DQS duty Cycle. No change for RD_DQS

NOTE

The above recommended configuration of DQS 4 -7 in MMDC2_MPDCCR is not required for x32 applications not using these byte lanes

Table 5. Summary of recommended DQS duty cycle fine tune parameter settings

MMDC1_MPDCCR (Register optimizations for DQS 0 - 3)	MMDC2_MPDCCR (Register optimizations for DQS 4 - 7)
WR_DQS0_FT_DCC = 001b	WR_DQS4_FT_DCC = 010b
WR_DQS1_FT_DCC = 001b	WR_DQS5_FT_DCC = 010b
WR_DQS2_FT_DCC = 010b	WR_DQS6_FT_DCC = 001b
WR_DQS3_FT_DCC = 010b	WR_DQS7_FT_DCC = 010b

Default Values of MMDC1_MPDCCR and MMDC2_MPDCCR registers after reset

```
0x021b08c0 = 0x24922492   // MMDC1_MPDCCR
0x021b48c0 = 0x24922492   // MMDC2_MPDCCR
```

Proposed Register Configuration for MMDC1_MPDCCR and MMDC2_MPDCCR

```
0x021b08c0 = 0x24912489 // Recommended MMDC1_MPDCCR setting
0x021b48c0 = 0x24914452 // Recommended MMDC2_MPDCCR setting
```

The recommended location for inserting the MMDC_MPDCCR registers settings for modifying the SDCLK duty cycle adjustments during the MMDC initialization would be immediately after the Program Calibration Setting Registers and before issuing the MMDCx_MPMUR0_FRC_MSR command to set calibration values in the PHY. An example is shown below:

```
// Duty Cycle Adjustment in DDR Initialization scripts
//====================================================================
     // Write Leveling calibration
// Read DQS Gating calibration
// Read calibration
// Write calibration
// SDCLK and DQS Duty Cycle adjustment
setmem /32       0x021b08c0 =     0x24912489
setmem /32       0x021b48c0 =     0x24914452
//====================================================================
// Complete calibration by forced measurement:
// MMDC initialization:
```

NOTE

The SDCLK Duty Cycle optimizations provided in this section are specific to i.MX 6DualPlus/6QuadPlus. For i.MX 6Dual/6Quad specific optimizations, see *SDCLK Duty Cycle Optimizations for i.MX 6Quad/6Dual* (document EB817).

4 VIX optimizations

VIX is defined by JEDEC as the delta voltage of the crossing point of the SDCLKx_B / SDQSx_B and SDCLKx / SDQSx signals versus the NVCC_DRAM / 2 voltage. The primary reasons for VIX variation in a system are as follows:

- Differential Output Phase Skew—timing difference of the differential negative and positive signals
- Output Slew Rate—difference in the slew rate of the differential signal
- Shifts in supply voltage levels—deviations in the voltage supply rail and/or the ground plane cause the VIX cross point to deviate from the defined Vref point (NVCC_DRAM / 2 voltage)
- Pulse Width—the DQS and the DQS_B have slightly different pulse widths (tDQSH/L) when comparing the high pulse of one signal to the low pulse of the other signal

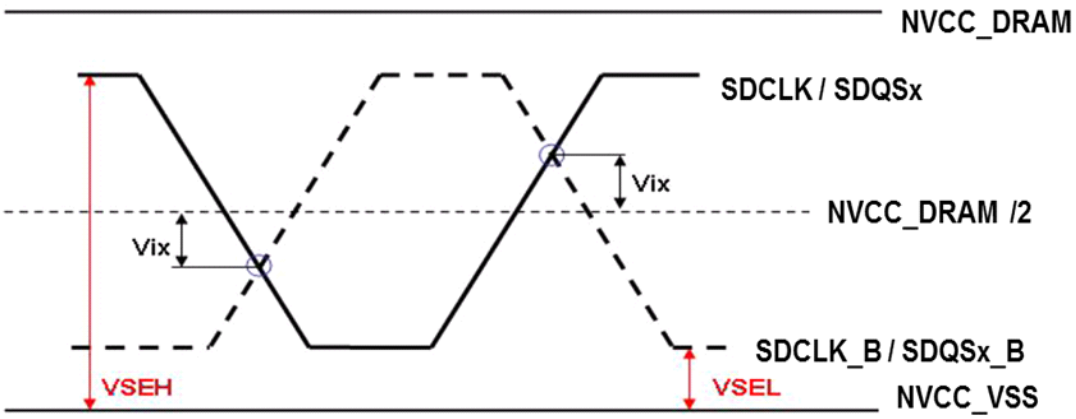

Figure 1. VIX parameter showing the crossing point of the SDCLK_B and SDCLK signals

Modifying the slew rate/ drive strength of the SDCLKx / SDQSx signals can impact the VIX measurement. The initial and periodic ZQ calibration configured and enabled during the DDR initialization ensures that the devices receive the correct calibration drive strength. The i.MX 6DualPlus/6QuadPlus MMDC has the new functionality added, which is a global offset register for the ZQ calibration PU (Pull Up) and PD (Pull Down) settings. This offset value can be automatically appended to the PU / PD value after ZQ calibration is completed, as an empirical adjustment to optimize the VIX crossing point.

This global offset is controlled by register MMDC_MPPDCMPR2 (Address: Base address + 890h offset) as stated in the i.MX 6DualPlus/6QuadPlus Reference Manual (Section 46.12.67). When the offset is enabled by the ZQ_OFFSET_EN bit, (MMDC_MPPDCMPR2[3]), the programmed offset value applies to all the MMDC differential signals (SDCLKx_B / SDQSx_B and SDCLKx / SDQSx).

Table 6. MMDC_MPPDCMPR2 register bit definitions required for the new ZQ offset feature

Register Bits	Setting	Notes
ZQ_PU_OFFSET MMDC_MPPDCMPR2 [11:8]	The ZQ pull up offset range is programmable from -7 to +7. Bit 11 controls the direction of the PU offset. Bits [10-8] control the PU offset value.	Only applied when ZQ_OFFSET_EN = 1 (Default register bits value out of reset = 0)
ZQ_PD_OFFSET MMDC_MPPDCMPR2 [7:4]	The ZQ pull down offset range is programmable from -7 to +7. Bit 7 controls the direction of the PD offset. Bits [6-4] control the PD offset value.	Only applied when ZQ_OFFSET_EN = 1 (Default register bits value out of reset = 0)
ZQ_OFFSET_EN MMDC_MPPDCMPR2[3]	0 = ZQ Offset Disabled 1 = ZQ Offset Enabled	(Default register bits value out of reset = 0 - disabled)

NOTE

Only 1 register exists to control the offset correction for both DDR3 and LPDDR2 dual channel modes of operation.

Design simulations combined with actual circuit measurements on NXP hardware, over worst case silicon process and temperature conditions were performed to derive the optimal settings for VIX performance. The testing utilized the latest DDR initialization scripts tuned for the NXP hardware.

The following settings, give the best margin on memory stress testing with Auto ZQ calibration enabled at the maximum supported 528 MHz DDR clock frequency.

NOTE

Variations due to the different board layouts, topology, memory selection, components and other application specific use case factors can yield slightly different results. Customers are recommended to optimize parameters for their end product appropriately.

Table 7. Register bit definitions required for new ZQ offset feature

Register Bits	Setting
ZQ_PU_OFFSET MMDC_MPPDCMPR2 [11:8]	0xC Bit [11] = 1b – (controls the direction of the PU offset) Bits [10-8] = 100b (PU offset value)
ZQ_PD_OFFSET MMDC_MPPDCMPR2 [7:4]	0x5 Bit [7] = 0b – (controls the direction of the PD offset). Bits [6-4] = 101b (PD offset value)
ZQ_OFFSET_EN MMDC_MPPDCMPR2[3]	0x1 Bit [3] = 1b (ZQ Offset Enabled)

Proposed MMDC_ MPPDCMPR2 Register Configuration

```
0x021b0890 = 0x00400C58 // MMDC_ MPPDCMPR2
```

The recommended location for inserting the MMDC_ MPPDCMPR2 registers settings for modifying the ZQ adjustments during the MMDC initialization would be before programming the ZQ Calibration Setting Registers and before issuing the MMDCx_MPMUR0_FRC_MSR command to set calibration values in the PHY.

```
// ZQ offset Adjustment in DDR Initialization scripts

//====================================================================

     // ZQ calibration global Offset


setmem /32       0x021b0890 =     0x00400C58


//====================================================================


// ZQ calibration

// Complete calibration by forced measurement:

// MMDC initialization:
```

4.1 DQS fine tuning

The DRAM controller has also been enhanced with additional trim controls for the SDCLKx and SDQSx signals. These new trim controls allow additional fine tuning to enable i.MX 6DualPlus/6QuadPlus to adhere to the JEDEC timing specification. Each of the differential pair signals has independent trim control, via the new IOMUX register and bit fields listed in section and Table 46-14. Additional DRAM trim controls in the i.MX 6DualPlus/6QuadPlus Applications Processor Reference Manual.

NOTE

The SDQS_x_TRIM control is a different delay than the READ fine tuning controlled by the MMDC register MPRDDQBY#DL.

The 2-bit control field for each signal is defined as shown in the following table. A single delay unit is typically defined as 30 ps and can be added separately to each pad of these differential pair signals. This delay does vary with silicon process, voltage and temperature.

Table 8. SDCLK_x_TRIM and SDQS_x_TRIM control definitions

Bit Field	Description	Notes
00	No additional delay	Default setting
01	1 delay unit added to pad edge	30 ps delay
10	2 delay units added to pad edge	60 ps delay
11	3 delay units added to pad edge	90 ps delay

Design simulations have shown adding the delay trim can help compensate for the phase shift and thereby improve the VIX performance. Actual circuit measurements were performed on NXP hardware, over

worst case silicon process and temperature conditions to derive the optimal settings for VIX performance. The testing utilized the latest DDR initialization scripts tuned for the NXP hardware.

The following settings, give the best VIX margin on memory stress testing at the maximum supported 528 MHz DDR clock frequency on NXP hardware. Users may further fine tune these settings for their production board design.

NOTE

Variations due to the different board layouts, topology, memory selection, components and other application specific use case factors can yield slightly different results. Customers are recommended to optimize parameters for their end product appropriately.

Table 9. SDQS_x_TRIM bit recommendations

Register Bits	Setting	Notes
IOMUXC_SW_PAD_CTL_PAD_DRAM_ADDR0x [9:8] IOMUXC_SW_PAD_CTL_PAD_DRAM_ADDR0x [17:16]	SDQSx_TRIM=10 (2 delay units) SDQSx_B_TRIM=01 (1 delay unit)	Delay setting applied to DQS0, DQS2 -7
IOMUXC_SW_PAD_CTL_PAD_DRAM_ADDR03 [9:8] IOMUXC_SW_PAD_CTL_PAD_DRAM_ADDR03 [17:16]	SDQS1_TRIM=00 (0 delay units) SDQS1_B_TRIM=00 (0 delay units)	Default value for DQS1 with no delay added

Proposed SDQS_x_TRIM bit configuration:

```
0x020e0534 =      0x00018200        // IOMUXC_SW_PAD_CTL_PAD_DRAM_ADDR02 (SDQS0_B_TRIM=01,
SDQS0_TRIM=10)

0x020e0538 =      0x00008000        // IOMUXC_SW_PAD_CTL_PAD_DRAM_ADDR03  (SDQS1_B_TRIM=00,
SDQS1_TRIM=00)

0x020e053C =      0x00018200        // IOMUXC_SW_PAD_CTL_PAD_DRAM_ADDR04  (SDQS2_B_TRIM=01,
SDQS2_TRIM=10)

0x020e0540 =      0x00018200        // IOMUXC_SW_PAD_CTL_PAD_DRAM_ADDR05  (SDQS3_B_TRIM=01,
SDQS3_TRIM=10)

0x020e0544 =      0x00018200        // IOMUXC_SW_PAD_CTL_PAD_DRAM_ADDR06  (SDQS4_B_TRIM=01,
SDQS4_TRIM=10)

0x020e0548 =      0x00018200        // IOMUXC_SW_PAD_CTL_PAD_DRAM_ADDR07  (SDQS5_B_TRIM=01,
SDQS5_TRIM=10)

0x020e054C =      0x00018200        // IOMUXC_SW_PAD_CTL_PAD_DRAM_ADDR08  (SDQS6_B_TRIM=01,
SDQS6_TRIM=10)

0x020e0550 =      0x00018200        // IOMUXC_SW_PAD_CTL_PAD_DRAM_ADDR09  (SDQS7_B_TRIM=01,
SDQS7_TRIM=10)
```

The recommended location for inserting the SDQS_x_TRIM registers settings for modifying the

DQS pad delay adjustments during the MMDC initialization would be during the initial IOMUX configuration (Step 2 of the sequence described in Section 2.1, “MMDC programming sequence”) and prior to step 3 (Programming the Calibration Setting Registers).

NOTE

NXP testing results indicated that adding TRIM delay on the SDCLK was not required. Customers can however similarly optimize VIX on SDCLK_x_TRIM if fine tuning is required on their production board design.

5 MMDC reordering and arbitration control

The i.MX 6DualPlus/6QuadPlus has some additional settings and configurability added to control the data reordering and arbitration control logic for improving DDR performance. The reordering optimizes the accesses and maximizes the utilization of the DDR bus. The MMDC Core AXI Reordering Control register (MMDCx_MAARCR) determines the values to be used for the reordering arbitration engine. See the MMDC chapter of the i.MX 6DualPlus/6QuadPlus reference manual (IMX6DQPRM) for more details on this feature.

Previously the MMDC_MAARCR register was not normally programmed by the DDR initialization script and NXP recommended leaving it with default out of reset values. However, on the i.MX 6DualPlus/6QuadPlus with improved bus fabric optimizations and addition of the NoC, we recommend some modifications the MMDC_MAARCR register. It is also important to ensure both the MMDC and the NoC are correctly configured to realize the optimal performance gains.

When the *ARCR_REO_DIS* bit (MMDC_MAARCR[25]) is set, the MMDC reorder pass-through mode is enabled. In this mode the DRAM access commands are still processed by MMDC reorder and arbitration pipeline, but the commands are not reordered. When the *ARCR_ARB_REO_DIS* bit (MMDC_MAARCR[26]) is set, the MMDC reorder and arbitration pipeline is totally bypassed. In this case, the DRAM access latency is reduced and performance is improved.

Table 10. MMDC_MAARCR Register bit definitions for Arbitration and Reordering

Register Bits	Settings	Notes
ARCR_REO_DIS MMDC_MAARCR[25]	0 - MMDC reordering controls enabled 1 - MMDC reordering controls disabled (bypassed)	MMDC reorder pass-through mode is enabled when this bit is set. When MMDCC_MAARCR[26] is set, MMDC_MARCR[25] has no impact since all AXI transactions bypass the reordering logic.
ARCR_ARB_REO_DIS MMDC_MAARCR[26]	0 - MMDC arbitration and reordering controls enabled 1 - MMDC arbitration and reordering controls disabled - Recommended	MMDC reorder and arbitration pipeline is totally bypassed when this bit is set. In this case the DRAM access latency is reduced and performance is improved

Recommended programming for MMDC_MAARCR assuming NOC is enabled is as follows:

```
setmem /32 0x021b0400 = 0x14420000      // adopt bypass - MMDC arbitration &
                                        // reordering controls disabled
```

NOTE

When the NoC Scheduler is enabled, the reordering function in the MMDC must be disabled.

NOTE

Due to errata *ERR009596* – MMDC ARCR_GUARD field in the MMDC_MAARCR register should not be reconfigured from the default value of 0. See the latest i.MX 6DualPlus/6QuadPlus Chip Errata document for more details.

6 NoC configuration

The NoC (by Arteris Ltd.) is a configurable high efficiency and performance fabric IP that was added to the i.MX 6DualPlus/6QuadPlus. See the Network Interconnect Bus System (NIC-301) chapter in the i.MX 6DualPlus/6QuadPlus reference manual (IMX6DQPRM) for more details. The majority of the design options are configured during hardware design integration stage. There are some application-specific configurations that are required to be configured by software and are briefly listed in this section.

On system boot up, software needs to set up the NoC internal Scheduler during the DDR initialization phase and should not be modified once the DDR memory is active. Dynamic configuration of these parameters can cause system instabilities.

To avoid potential conflict, the corresponding bus controls must match these new MMDC settings. The bus controls are defined by the BOOT_CFG3[5:4] and BOOT_CFG3[1:0] bits. See the Fuse Map or for more details.

Table 11. eFUSE configuration of BOOT_CFG3 for NoC configuration

WorkMode = {BOOT_CFG3[1:0], BOOT_CFG3[5:4]}	DDR Memory Map Configuration	NoC Scheduler	MMDC Reordering (Required MMDC_MAARCR setting)
4'bxx00	Single DDR channel	Enabled [1]	Disabled [2]
4'bxx01	Fixed 2x32 map	Disabled	Enabled
4'bxx10	4KB Interleaving	Disabled	Enabled
4'b0011	Single DDR channel	Disabled	Enabled
4'b0111	Fixed 2x32 map	Enabled [1]	Disabled [2]

1. The NoC Scheduler is capable of optimizing the DRAM access pattern hence increasing the overall DRAM efficiency. Therefore it is highly recommended to choose NoC Scheduler enabled work modes (4'bxx00) for better performance if possible.
2. When the NoC Scheduler is enabled, the reordering function in the MMDC must be disabled.

6.1 NoC register configuration

The NIC-301 and NoC default settings are configured by NXP's board support package (BSP), and in most cases should not be modified by the customer. The default settings have gone through exhaustive testing during the validation of the device, and have proven to work well for the part's intended target applications.

The following NoC registers are fixed based on integration parameters and should not be modified by users:

- DDR Mode
- Read Latency
- AgingIPU1/IPU2

Changes to the above settings may result in degradation in system performance. For details on these registers, refer to the NoC section in the Network Interconnect Bus System (NIC-301) chapter in the i.MX 6DualPlus/6QuadPlus reference manual (IMX6DQPRM).

The following NoC registers are dependent on DDR timing parameters and will require modification based on the customer's selected DDR3 memory device:

- DDRTIMING
- DDRCONF
- ACTIVATE

Because the above parameters are dependent on DDR timing parameters, we recommend customers use the DDR3 MMDC Excel register programming aid to assist in configuring these specific parameters. Contact your NXP applications field engineer or sales representative for this programming aid.

The following section briefly describes these NoC registers and provides an example configuration based on the Micron DDR3L memory (MT41K256M16HA-125:E) used on the NXP i.MX6DualPlus/6QuadPlus Auto Infotainment Reference Board. If customers are using the same exact memory and configuration as the NXP board in their product design, then values provided by NXP can be used without further modification.

6.2 NoC DDRTIMING

Parameters for the NoC DDRTIMING register (0x00BB 000C) are derived from certain fields of the MMDC_MDCFG0/1/2 registers. Users must therefore refer to the DDR3 memory datasheet to obtain the specific parameters that need to be programmed in this register.

Table 12. NoC DDRTIMING register bit field descriptions

NoC DDR Timing Register bits	Bit offset	Description
BWRATIO	31	BwRatio: Number of cycle minus 1 the DDR chip needs to process one generic socket word.
WRTORD	26	WrToRd: Minimum number of DRAM clock cycles between the last DRAM Write command and a Read command. (tCWL+tWTR)
RDTOWR	21	RdToWr: Minimum number of scheduler DRAM clock cycles between the last DRAM Read command and a Write command. (tCL-tCWL+2)
BURSTLEN	18	BurstLen: DRAM burst duration on the DRAM data bus in scheduler DRAM clock cycles. Also equal to scheduler clock cycles between two DRAM commands. "4" if BL8, "2" if BL4
WRTOMISS	12	WrToMiss: Minimum number of DRAM clock cycles between the last DRAM Write command and a new Read or Write command in another page of the same bank. (tCWL+tWR+tRP+tRCD)
RDTOMISS	6	RdToMiss: Minimum number of DRAM clock cycles between the last DRAM Read command and a new Read or Write command in another page of the same bank. (tRTP+tRP+tRCD-BL/2)
ACTTOACT	0	ActToAct : Minimum number of DRAM clock cycles between two consecutive DRAM Activate commands on the same bank.(tRC)

6.3 NoC ACTIVATE

Parameters for the NoC ACTIVATE register (0x00BB 0038) are also derived from certain fields of the MMDC_MDCFG0/1/2 registers. Users must refer to the DDR3 memory datasheet to obtain the specific parameters that need to be programmed in this register.

Table 13. NoC ACTIVATE register bit field descriptions

NoC Activate Register bits	Bit Offset	Description
RD	0	Number of cycles between two consecutive Activate commands on different banks of the same device.
FAW	4	Number of cycles of the FAW period.
FAW BANK	10	Number of bank of a given device involved in the FAW period.

6.4 NoC configuration example

The following table illustrates how the correct NoC parameters for the NoC DDRTIMING and ACTIVATE registers can be derived, based on the timing parameters of the Micron DDR3L memory (MT41K256M16HA-125:E) used on the NXP i.MX 6DualPlus/6QuadPlus Auto Infotainment Reference Board.

Table 14. NoC Configuration Settings Example

MMDC configuration	Input	Registers	(HEX)
		MMDC.MDCTL	841A0000
		MMDC.MDCFG0	898E7955
		MMDC.MDCFG1	FF320F64
		MMDC.MDCFG2	01FF00DB
Mapped NoC configuration	**Output**	**Registers**	**(HEX)**
		NoC.DDRTIMING	2891E41A
		NoC.ACTIVATE	564

Since the above parameters are dependent on DDR memory timing parameters, we recommend customers use the DDR3 MMDC Excel register programming aid to assist in configuring these specific parameters.

6.5 NoC DDRCONF

The NoC DDRCONF register (0x00BB 0008) is used to select predefined memory configurations. This register allows the NOC to be optimized for the specific row/column/back architecture of the DRAM device being used. Since the various JEDEC compatible memory configurations are already encoded, users just need to select the correct configuration. See the following tables for the NoC predefined configuration settings.

Table 15. Example of possible DDRCONF settings for various 64 bit DDR3 memories

DDR3	8-banks, 64-bit						
	DDR Rows:						
		11	12	13	14	15	16
DDR Columns:	8						
	9						
	10		0	0	0	**4** (2 Chip Selects) **0** (1 Chip Select)	0
	11			1	1	1	
	12						

NOTE

Shaded cells correspond to non-compliant JEDEC row/column combinations.

For most common 64 bit DDR3 memory configurations with bank interleaving enabled, the DDRCONF value of 0 is the correct value that should be programmed during the DDR initialization.

Table 16. Example of possible DDRCONF settings for various 32 bit DDR3 memories

DDR3	8-banks, 32-bit						
	DDR Rows:						
		11	12	13	14	15	16
DDR Columns:	8						
	9			**2**	15		
	10		3	3	**14** (2 Chip Selects) **3** (1 Chip Select)	**9** (2 Chip Selects) **3** (1 Chip Select)	3
	11			0	0	**4** (2 Chip Selects) **0** (1 Chip Select)	0
	12				**1**		

NOTE

Shaded cells correspond to non compliant JEDEC row/column combinations

6.6 NoC initialization register settings

As described in Section 2, “MMDC register programming”, the NoC programming is performed towards the end of the DDR initialization sequence as long as it is before the clearing of the Configuration Request *(con_req)* bit in MMDC0_MDSCR.

```
// NoC initialization
//====================================================================
setmem /32 0x00bb0008 = 0x00000000   ## NoC.DDRCONF Row= 15bits, Col= 10bits, 64bit
setmem /32 0x00bb000c = 0x2891E41A   ## NoC.DDRTIMING according to MMDC0_MDCFG0/1/2
setmem /32 0x00bb0038 = 0x00000564   ## NoC.ACTIVATE according to MMDC0_MDCFG0/1/2
setmem /32 0x00bb0014 = 0x00000040   ## NoC.ReadLatency (Fixed)
setmem /32 0x00bb0028 = 0x00000020   ## NoC.AgingIPU1 Aging control-IPU1/PRE0/PRE3
setmem /32 0x00bb002c = 0x00000020   ## NoC.AgingIPU2 Aging control-IPU2/PRE1/PRE2
//====================================================================


// MMDC and NoC initialization complete


// Clear Configuration Request (con_req) bit in MMDC0_MDSCR.
```

7 General recommendations to improve performance and JEDEC compliance

The following section provides brief recommendations on how to further optimize the DDR interface. The following recommendations are not absolutely necessary to optimize various parameters and may also have limited applicability on certain customer designs.

7.1 Drive strength configuration

NXP testing has shown that lowering the drive strengths can degrade the SDCLK Duty Cycle performance. NXP does not recommend using 60 Ohms or lower for any customer design. For the IOPAD Drive Strength, users should set the DSE = 110b (40 Ohms (default value) bits 5, 4, 3) in the IOMUXC_SW_PAD_CTL_PAD_DRAM_XX registers.

NXP testing has also shown that using DSE = 110b (40 Ohms (default value) bits 5, 4, 3) provides optimal VIX and tDSQH performance. Note that drive strength configuration and optimization is dependent on the customer board layout and design.

7.2 Supply configurations

NXP testing has shown that increasing the VDD_SOC_CAP voltage level from the defaults specification can improve the SDCLK Jitter and VIX marginally. Increasing VDD_HIGH_CAP to 2.65 V can also help improve jitter and VIX performance.

Example register programming to increase the VDD_HIGH_CAP from the default voltage of 2.5 V is shown below:

```
0x020C8130 0x00001073  <== Default value for 2.50v PMU_REG_2P5

0x020C8130 0x00001473  <== Change for 2.60v PMU_REG_2P5

0x020C8130 0x00001673  <== Change for 2.65v PMU_REG_2P5
```

The VDD_SOC_CAP voltage level setting can be modified in the PMU_REG_CORE register in the Power Management Unit (PMU). Note that the VDD_SOC_CAP and VDD_HIGH_CAP optimization is dependent on the customer design and the operating ranges defined in the respective i.MX 6DualPlus/6QuadPlus data sheets. Depending on the customer design and power restrictions it may not be possible to modify the VDD_SOC_CAP voltage level.

7.3 Supply ripple

Regulation instabilities and ripples on the output of the LDO can also increase the system clock jitter which can impact the SDCLK Duty Cycle and other DDR signal parameters. Customers should specifically ensure correct capacitors sizing and placement on NVCC_PLL_OUT, VDD_SOC_CAP, VDD_HIGH_CAP and NVCC_DRAM.

To minimize jitter, follow the layout and decoupling recommendations in the i.MX 6 Hardware Development Guide for the i.MX 6 series.

8 References

1. *JEDEC DDR3 SDRAM Standard JESD79-3 F* (document JESD79-3 F)
2. *i.MX6 Hardware Development Guide for i.MX 6 Families of Applications Processors* (document IMX6DQ6SDLHDG)
3. *i.MX 6DualPlus/6QuadPlus - segment Data Sheets* (documents IMX6DQPCEC, IMX6DQPAEC, and IMX6DQPIEC)
4. *i.MX 6DualPlus/6QuadPlus Applications Processor Reference Manual* (document IMX6DQPRM)
5. *i.MX 6DualPlus/6QuadPlus Applications Processor Chip Errata* (document IMX6DQCE)
6. *i.MX 6DualPlus/6QuadPlus DDR3 Register Programming Aid*
7. *i.MX 6Dual/6Quad to i.MX 6DualPlus/6QuadPlus Comparison* (document EB810)
8. *SDCLK Duty Cycle Optimizations for i.MX 6Quad/6Dual* (document EB817)

9 Revision history

The following table summarizes revisions to this document since the release of the previous version.

Table 17. Revision history

Revision	Date	Substantive changes
0	08/2016	• Added further configuration optimizations for VIX in Section 4.1, “DQS fine tuning”. • Added further configuration updates for tDQSH/L. • Updated general recommendations to improve performance and JEDEC compliance.

How to Reach Us:

Home Page:
nxp.com

Web Support:
nxp.com/support

Document Number: EB828
Rev. 0
08/2016