Freescale Semiconductor
Application Note

Document Number: AN2583
Rev. 11, 07/2014

Programming the PowerQUICC III/PowerQUICC II Pro DDR SDRAM Controller

The fully programmable DDR-SDRAM memory controller supports JEDEC-compatible DDR1 SDRAM memories up to 166 MHz (333 MHz data rate). This application note provides programming guidelines for the PowerQUICC DDR-SDRAM memory controller and specifically JEDEC-compatible DDR1 SDRAM memories.

Contents

1. DDR1 SDRAM Overview 2
2. Software Considerations 3
3. DDR-SDRAM Example 7
4. Source Synchronous Mode 9
5. Documentation References 22
6. Document Revision History 22

The Freescale PowerQUICC III and PowerQUICC II Pro devices are the latest PowerQUICC integrated communication processors and the first PowerQUICC family members to include an on-chip DDR-SDRAM memory controller. For a list of all PowerQUICC III devices, see the website listed on the back cover of this document. Table 1 lists the devices covered in this application note, which are collectively referred to in the rest of this paper as the "PowerQUICC devices."

Table 1. PowerQUICC III and PowerQUICC II Pro Families

PowerQUICC III	PowerQUICC II Pro
MPC8540	MPC8349
MPC8560	MPC8347
MPC8555E[1]	MPC8343
MPC8541E [1]	—

[1] These devices are sometimes collectively referred to as PowerQUICC III Lite.

1 DDR1 SDRAM Overview

During the 1990s, the PC market shifted towards faster CPUs and improved performance. To take advantage of this shift in the market, memory manufacturers defined a new type of SDRAM technology that supports higher memory bandwidths. Double data rate (DDR) memory is the next-generation SDRAM, and like its predecessor, DDR is synchronized with the system clock. DDR provides notable product enhancements, including increased performance at reduced power consumption levels. Table 2 summarizes the main differences between DDR1 and SDRAM.

Table 2. Differences Between DDR-I and SDRAM

Summary	Description
Data rates	DDR memory provides source-synchronous data capture at a rate of twice the clock frequency (that is, data is transferred on both edges of the clock). This enhancement allows the DDR module to transfer data twice as fast as SDRAM. For example, instead of a data rate of 133 MHz, DDR memory transfers data at 266 MHz.
Digital local loops (DLLs)	DDR memory has on-chip digital lock loops (DLLs) to minimize clock skew.
Memory access	DDR memory is burst-oriented memory. Single accesses are not allowed.
DQM functionality	DQM functionality of SDRAM memories is replaced with data masks (DMs) in addition to the introduction of bi-directional Data Strobe DQS for source synchronous data capture
IO logic	DDR-I memory IO logic is SSTL_2 Series Stub Terminated Logic 2.5 V.

Many different DDR-SDRAM configurations are supported, including buffered and unbuffered standard DIMM modules or directly attached memory devices. However, unbuffered and registered DIMMs cannot be mixed in the same system.

Fourteen multiplexed address signals and two logical bank signals provide support for device densities of 64 Mbits to 1-Gbit. Four chip select signals support up to four physical banks of memory, each from 64 MBytes to 1 GByte in size or up to two DIMM modules. While these four banks provide up to a maximum of 4 GBytes of DDR main memory, the 4 GBytes would span the entire 32-bit address space. However,

because space must be reserved for boot ROM, configuration registers, and other important addressable locations, the maximum DDR memory is limited to 3.5 GBytes.

The PowerQUICC devices can be configured to retain the currently active SDRAM page for pipelined burst accesses. Page mode support of up to 16 simultaneously open pages (4 for each memory bank) can dramatically reduce access latencies for page hits. Read and write accesses to the DDR SDRAM are burst-oriented; accesses start at a selected location and continue in a sequential or interleaved sequence. (The DDR controller supports only burst-of-four transfers.) Optional error checking and correcting (ECC) protection are provided for the DDR SDRAM data bus. Using ECC, the DDR memory controller detects and corrects all single-bit errors within the 64-bit data bus, detects all double-bit errors within the 64-bit data bus, and detects all errors within a nibble.

With the advancements in processor speeds, as well as those in memory technology, the next generation of DDR memory devices (DDR2) sampling now offers further performance improvements at reduced power consumption levels. Future PowerQUICC derivatives (in 90nm HiP8 technology) will include an integrated fully programmable DDR-SDRAM memory controller that supports JEDEC-compatible DDR2 SDRAM memories.

2 Software Considerations

The following sections discuss software considerations such as the local memory map and DDR controller.

2.1 Local Memory Map

The PowerQUICC and PowerQUICC II Pro III have a flexible local memory map that refers to the 32-bit, 4-GByte address space used to access DDR SDRAM, local bus memory controllers, all configuration, control, and status registers (CCSR), and I/O space. The local memory map is defined by a set of eight local access windows (LAWs), each of which maps a region of memory to a particular target interface such as the DDR SDRAM controller or the PCI controller. The size of each LAW can be configured from 4 KBytes to 2 GBytes.

Except for the configuration space (mapped by CCSRBAR), on-chip SRAM regions (mapped by L2SRBAR registers), and default boot ROM, all addresses used in a system must be mapped by a LAW. The default location for the boot ROM is the highest 8 MBytes of local memory from address 0xFF80_0000 to 0xFFFF_FFFF. The location for CCSR is programmable using the base register (CCSRBAR). The local memory map covered in this document is shown in Figure 1. This map is configured using U-Boot as a debug monitor on the MPC8560 application development system (ADS) pilot board. The DDR SDRAM is mapped into the lowest quarter of the local memory map.

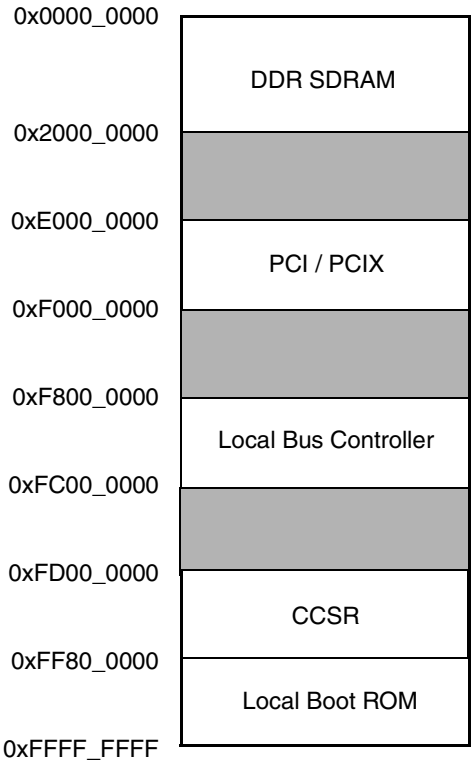


Figure 1. Local Memory Map for MPC8560ADS Pilot Board

The CCSRBAR and LAW register setting for the local memory map configured using U-Boot as a debug monitor on the MPC8560 ADS pilot board are described in Table 3.

Table 3. CCSRBAR and LAW*n* Register Settings

Name	Offset[1]	Description	Value
CCSRBAR	0x0_0000	Configuration control status base address register. Defines 12 most-significant address bits of the base of window for configuration accesses	0x000F_D000
LAWBAR1	0x0_0c28	Local access window 1 base address register. Defines 20 most-significant address bits of the base of local access window 0	0x0000_0000
LAWAR1	0x0_0c30	Local access window 1 attribute register • EN = 1 (enables local access window 0), • TRGT_IF = 1111 (identifies DDR SDRAM as the target interface) • SIZE = 01_1010 (identifies a 128-MByte window size)	0x80F0_001A
LAWBAR2	0x0_0c48	Local access window 2 base address register. Defines 20 most-significant address bits of the base of local access window 1	0x000E_0000
LAWAR2	0x0_0c50	Local access window 2 attribute register • EN = 1 (enables local access window 1), • TRGT_IF = 0000 (identifies PCI as the target interface) • SIZE = 01_1011 (identifies a 256-MByte window size)	0x8000_001B

Table 3. CCSRBAR and LAW*n* Register Settings (continued)

Name	Offset[1]	Description	Value
LAWBAR3	0x0_0c48	Local access window 3 base address register. Defines 20 most-significant address bits of the base of local access window 1	0x000F_8000
LAWAR3	0x0_0c50	Local access window 3 attribute register • EN = 1 (enables local access window 1), • TRGT_IF = 0100 (identifies local bus as the target interface) • SIZE = 01_1001 (identifies a 64-MByte window size)	0x8040_0019

1 Offset from the configuration, control and status registers (CCSR) base address

2.2 DDR Controller—Programming Model

The main registers for programming the internal PowerQUICC DDR1 memory controller are listed in Table 4.

Table 4. DDR SDRAM Memory Controller Registers

Offset[1]	Register
0x0_2000 - 18	Memory bounds registers for chip selects 0 to 3 (CS0_BNDS to CS3_BNDS)
0x0_2080 - 8C	Configuration registers for chip selects 0 to 3 (CS0_CONFIG to CS3_CONFIG)
0x0_2108	DDR SDRAM timing configuration 1 (TIMING_CFG_1)
0x0_210C	DDR SDRAM timing configuration 2 (TIMING_CFG_2)
0x0_2110	DDR SDRAM control configuration (DDR_SDRAM_CFG)
0x0_2118	DDR SDRAM mode configuration (DDR_SDRAM_MODE)
0x0_2124	DDR SDRAM interval configuration (DDR_SDRAM_INTERVAL)

1 Offset from the configuration, control and status registers (CCSR) base address

To configure the DDR memory controller, software must initialize the appropriate parameters in the following registers:

- Chip Select memory bound registers (CSn_BNDS), which define the address range for memory banks controlled by that Chip Select
- Chip Select configuration registers (CSn_CONFIG), which define the memory organization (number of row/columns)
- DDR SDRAM timing configuration registers (TIMING_CFG1 and TIMING_CFG2) to define timing intervals between various SDRAM control commands (see Table 5)
- DDR SDRAM configuration register (DDR_SDRAM_CFG) to enable or disable features of the memory controller, including self-refresh, ECC, and dynamic power management
- DDR SDRAM mode configuration (DDR_SDRAM_Mode) to define the mode registers of the SDRAM device
- DDR SDRAM interval configuration (DDR_SDRAM_INTERVAL) to configure the precharge and refresh intervals (see Table 5)

Table 5 indicates the DDR SDRAM timing intervals that can be programmed in the memory controller.

Table 5. Summary of DDR SDRAM Timing Intervals

Timing Intervals[1]	Definition	Register
ACTTOACT	Number of memory clock cycles from an active command to a logical bank until another active command is allowed for any logical bank within that bank	TIMING_CFG1
ACTOPRE	Number of memory clock cycles from an activate command until a precharge command is allowed	TIMING_CFG1
ACTORW	Number of memory clock cycles (t_{rcd}) from an activate command until a read or write command is allowed	TIMING_CFG1
BSTOPRE	Number of memory clock cycles to maintain a page open after an access. A subsequent access can generate a page hit during this interval. A page hit reloads the BSTOPRE counter. When the interval expires, a precharge is issued to the page as soon as possible.	SDRAM_INTERVAL
CASLAT	READ latency (CASLAT) is the delay, in memory clock cycles, between the registration of a READ command by the SDRAM and the availability of the first piece of output data. If a READ command is registered at clock edge n and the latency is m clocks, the data is available nominally coincident with clock edge $n + m$.	TIMING_CFG1
PRETOACT	Number of memory clock cycles (t_{rp}) from a precharge command until an activate or a refresh command is allowed.	TIMING_CFG1
REFINT	Refresh interval. Represents the number of memory bus clock cycles between refresh cycles. One row is refreshed in each SDRAM bank during each refresh cycle. The value of REFINT depends on the specific SDRAMs used and the frequency of the interface.	SDRAM_INTERVAL
REFREC	The number of memory clock cycles (t_{rfc}) from the refresh command until an activate command is allowed.	TIMING_CFG1
WR_DATA_DELAY	Provides different options for the timing between a write command and the write data strobe, allowing write data to be sent later than the nominal time to meet the SDRAM timing requirement between the registration of a write command and the reception of a data strobe associated with the write command. The specification dictates that the data strobe may not be received earlier than 75% of a cycle or later than 125% of a cycle, from the registration of a write command. This parameter is not defined in the SDRAM specification. It is implementation-specific, defined for the DDR memory controller in TIMING_CFG_2.	TIMING_CFG2
WRREC	The number of memory clock cycles (t_{wr}) from the last beat of a write until a precharge command is allowed.	TIMING_CFG1
WRTORD	Last write pair to read command issue for DDR and DDRII. Controls the number of memory clock cycles (t_{wtr}) from the last write data pair to the subsequent read command to the same bank.	TIMING_CFG1

[1] These timing intervals are documented in the AC characteristics of the DDR SDRAM discrete memory device or DIMM module.

When all the parameters are configured, system software must enable the DDR memory and initialize the DDR SDRAM devices according to the JEDEC standard as follows:

1. Wait 200 µs after the DLL is locked. After 200 µs has elapsed, software should enable the DDR memory controller by setting the MEM_EN bit in the DDR_SDRAM_CFG register.
2. Precharge all banks.

3. Mode register set for extended mode register (in the device)
4. Mode register set for mode register (in the device) with reset DLL bit set
5. Precharge all banks
6. Two Auto Refresh commands
7. Mode register set for mode register (in the device) with reset DLL bit cleared

The PowerQUICC III/PowerQUICC II Pro DDR memory controller automatically performs steps 2 to 7 after it has been enabled in step 1.

3 DDR-SDRAM Example

The example discussed in this section is based on the MPC8560ADS pilot board (see Figure 2). The 128-MBytes of DDR SDRAM memory is configured using one Micron MT9VDDT1672A 128-MByte unbuffered DIMM module connected to the board through a 184-Pin DIMM socket.

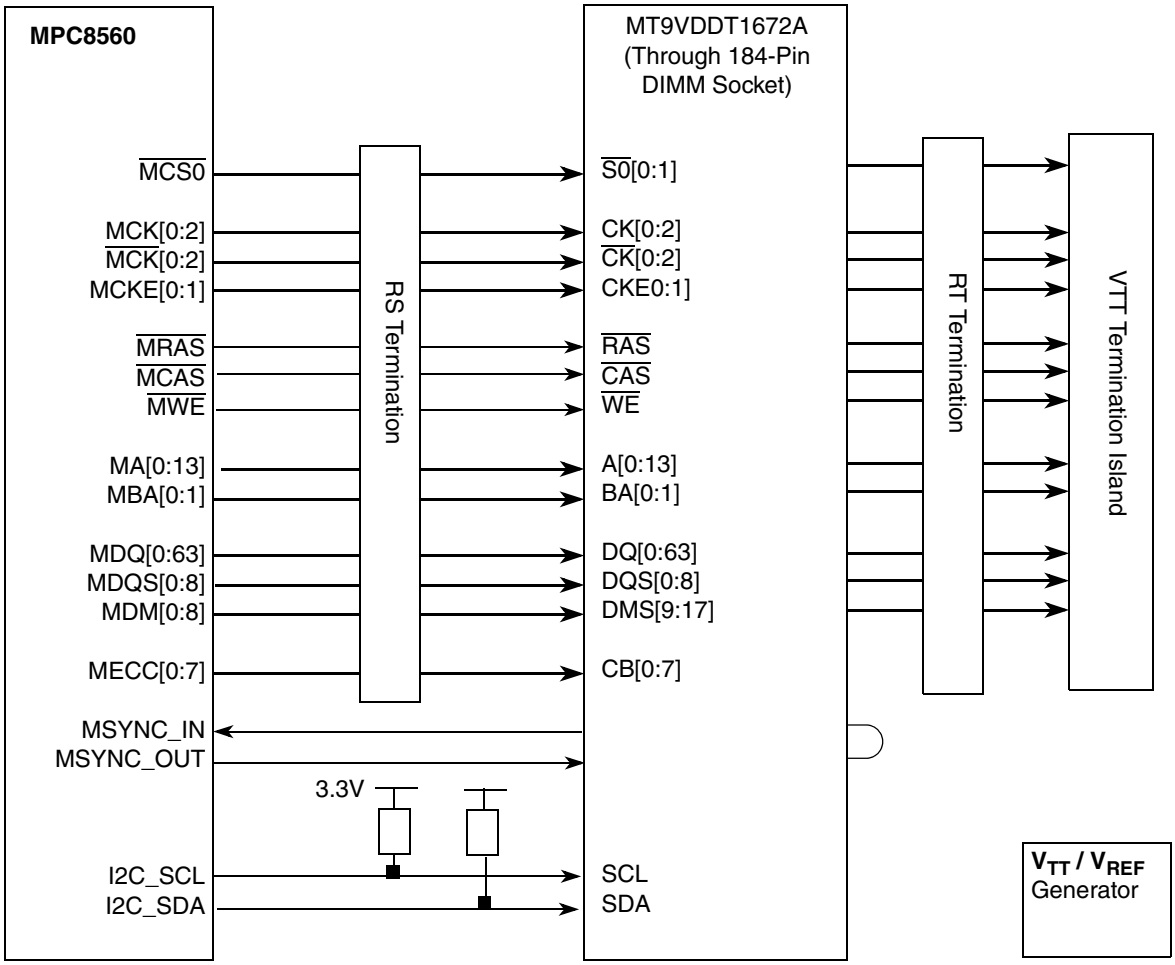


Figure 2. DDR SDRAM DIMM Connection on MPC8560ADS Pilot Board

At system reset, initialization software must configure the programmable parameters in the memory interface configuration registers. The sequence of events to initialize the DDR controller and the DDR SDRAM array after HRESET negation is as follows:

1. Disable the memory interface with SDRAM_CFG[MEM_EN].
2. Configure the appropriate DDR controller registers (for this example refer to Table 6).
3. Wait until the remainder of 200 µs has elapsed (the required time before the memory interface is enabled).
4. Enable the memory interface with SDRAM_CFG[MEM_EN].

Table 6 shows DDR controller register settings in the example.

Table 6. Example DDR Controller Register Settings

Name	Offset[1]	Description	Value
CS0_BNDS	0x0_2000	Chip select 0 memory bound register • SA0 = 0000_0000 “indicates the most significant 8 bits of the bank starting address” • EA0 = 0000_0111 “indicates the most significant 8 bits of the bank ending address”	0x0000_0007
CS0_CONFIG	0x0_2080	Chip select 0 configuration register • CS_0_EN = 1 “Enables bank 0 with the memory bounds described in CS0_BNDS” • AP_0_EN = 1 “Enable auto precharge mode of operation” • ROW_BITS_CS_0 = 000 “Set the number of row bits to 12” • COL_BITS_CS_0 = 010 “Set the number of column bits to 10”	0x8080_0002
TIMING_CONFG_1	0x0_2108	DDR SDRAM timing configuration 1 register • PRETOACT = 011 “Set the precharge to activate interval (t_{rp}) to 3 clock cycles” • ACTTOPRE = 111 “Set the activate to precharge interval (t_{ras}) to 7 clock cycles” • ACTTORW = 101 “Set the activate to read/write interval for SDRAM (t_{rcd}) to 5 clock cycles” • CASLAT = 100 “Set the $\overline{CAS}$ latency for reads to 2.5 clock cycles” • REFREC = 0100 “Set the refresh recovery time (t_{rfc}) to 12 clock cycles” • WRREC = 11 “Set the data to precharge interval (t_{wr}) to 3 clock cycles” • ACTTOACT = 010 “Set the activate to activate interval (t_{rrd}) to 2 clock cycles” • WRTORD = 01 “Set the write data to read command interval (t_{wtr}) to 1 clock cycle”	0x3754_4321
TIMING_CONFG_2	0x0_210C	DDR SDRAM timing configuration 1 register • WR_DATA_DELAY = 010 “Set delay applied to the data and data strobes for writes to 4/8 clock delay”	0x0000_0800
DDR_SDRAM_CFG	0x0_2110	DDR SDRAM mode configuration register • MEM_EN =1 “Enables DDR SDRAM memory controller” • SREN = 1 “Enables self refresh during sleep” • ECC_EN = 0 “Disable ECC interrupt generation” • RD_EN = 0 “Indicate unbuffered DIMMs” • SDRAM_TYPE = 10 “Indicates DDR SDRAM” • DYN_PWR = 0 “Disables dynamic power management mode”	0xC200_0000

Table 6. Example DDR Controller Register Settings (continued)

Name	Offset[1]	Description	Value
DDR_SDRAM_MODE	0x0_2118	DDR SDRAM mode configuration register • ESDMODE = 0000_0000_0000_0000 “Indicates initial value loaded into DDR SDRAM Extended mode register (normal drive strength and DLL enabled as specified in the DIMM module data sheet” • SDMODE = 0000_0000_0110_0010 “Indicates initial value loaded into DDR SDRAM base mode register (normal operation, $\overline{\text{CAS}}$ latency of 2.5, and sequential bursts of four as specified in the DIMM module data sheet.”	0x0000_0062
DDR_SDRAM_INTERVAL	0x0_2124	DDR SDRAM interval configuration register • REFINT = 00_0011_1110_1000 “Specify a refresh interval in clock cycles; 6 µs is specified, but if 6 µs is used to account for buffering of ongoing operations,6 µs / 6 ns cycle = 1000.” • BSTOPRE = 00_0000_0000_0000 “Indicates that the DDR memory controller uses auto precharge read and write commands rather than operating in page mode.”	0x03E8_0000

[1] Offset from the Configuration, Control and Status Registers (CCSR) base address

4 Source Synchronous Mode

The DDR controller is operated in source synchronous mode, which is selectable only in the MPC8555E, MPC8541E, and MPC834x rev 1.x when the internal DLL is not enabled. This mode is selected by setting the SS_EN bit in the DDR_SDRAM_CLK_CNTRL register. See Figure 3. However, in the MPC8540 and the MPC8560, the DLL continues to provide MCKs in override mode, but with a pre-programmed phase relation between MCK and the command bus, which is similar to what occurs in source clock mode.

If MSYNC_out board delay = MCK board delay, then MCK phase at DDR SDRAM = Msync_in at PowerQUICC III/PowerQUICC II Pro.

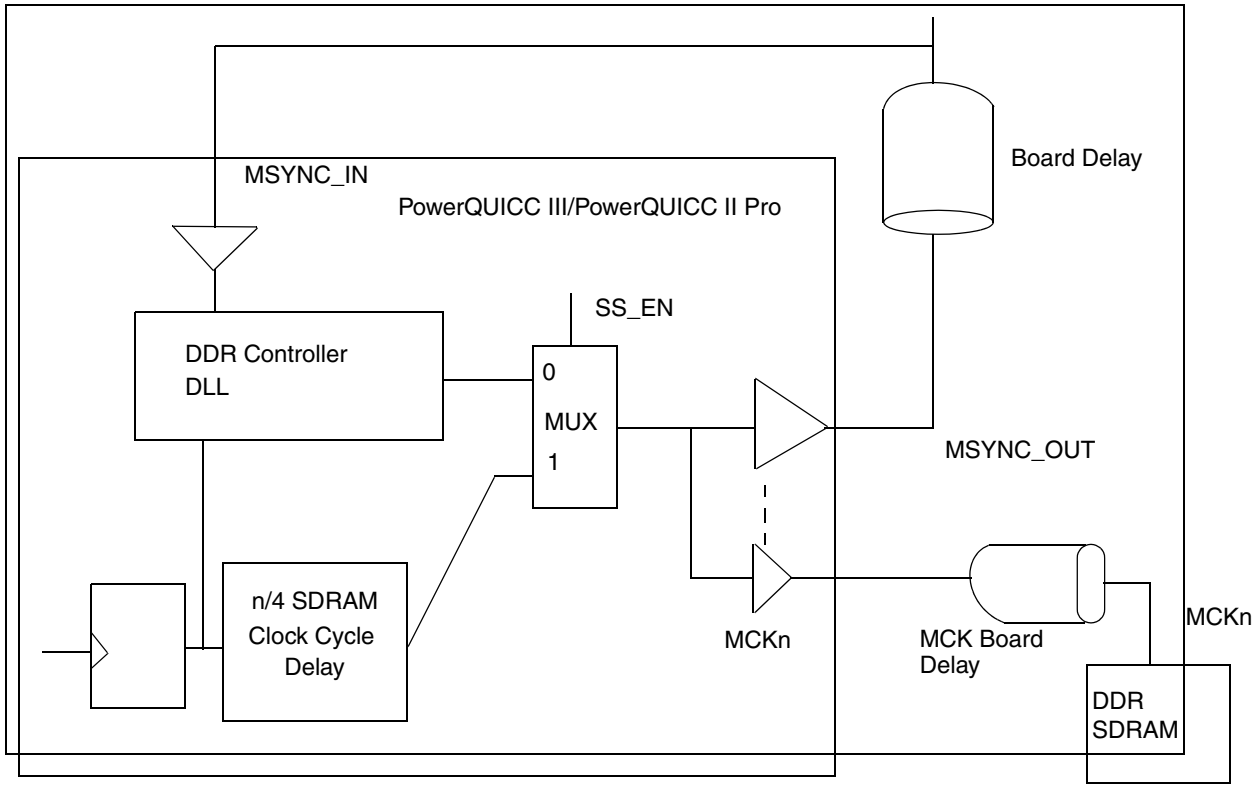


Figure 3. DLL and Source Clock Modes in MPC8555E, MPC8541E, and MPC834x rev 1.x

In DLL mode, the following occurs:

- MCKs are generated from the DLL and phase-aligned with MSYNC_OUT.
- MSYNC_IN is phase-aligned with the internal clock that is used to launch the command.

If board delays of MCKs and MSYNC_OUT are identical, the DLL guarantees that the PowerQUICC III/PowerQUICC II Pro internal clock used to launch commands is phase-aligned with the clock seen at the DDR SDRAM device. The DLL compensates all voltage- and temperature-delay related variations of MCK through chip IO buffers. Therefore, the relative phase of MCK is fixed at the DDR SDRAM device, but the phase alignments of command signals are allowed to vary with voltage and temperature relative to MCK.

However, in source clock mode, MCKs and command IO buffer delays are not compensated across possible variations of voltage and temperature. If the original phase relation between MCK and the command is known, the phase relation is maintained at the DDR SDRAM device. Thus, the command signal eye diagram for source clock mode is wider than the command signal eye diagram for DLL mode. Figure 4 illustrates this difference between the phase relationship in DLL and source clock modes. For each mode, the phase relationship between MCK and the command signal is represented by the distance between two pendulums.

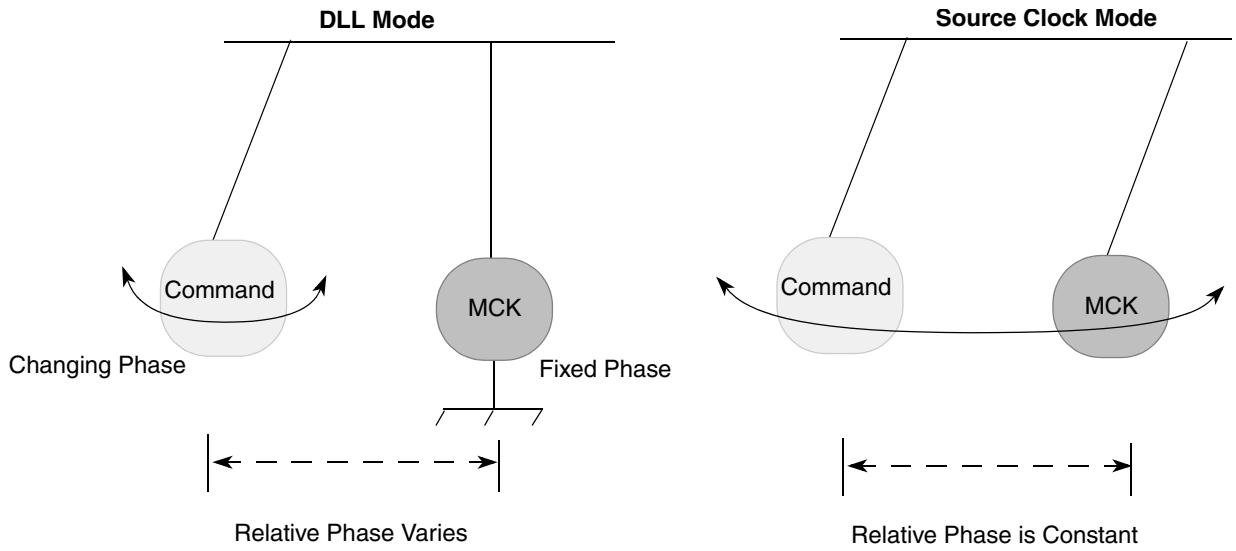


Figure 4. Phase Variation Due to Voltage, Temperature, And So On

See application note *Hardware and Layout Design Considerations for DDR Interfaces* (AN2582) for more detailed numerical figures that show the merit of using source clock synchronous mode. To configure the DDR controller in source synchronous mode, the programmer must evaluate two register field values:

- MCK CLK_ADJ—MCK clock adjustment
- CPO—MCAS to preamble over-ride

Section 4.1, "The $\overline{\text{DBWO}}$ Signal," describes MCK CLK_ADJ. The computed adjustment value of MCK can then be used for the CPO value computation, as discussed in sections 4.2 and 4.2.1.

4.1 MCK Adjustment

The adjustment of MCK is necessary to compensate for the extra incurred delays of the commanded bus due to loading as well as board routing length differences. MCK adjustment is a field in the PowerQUICC III Lite and PowerQUICC II Pro DDR SDRAM Clock Control Register.

4.1.1 MCK Adjustment for MPC8555E, MPC8541E, MPC8349, MPC8347, and MPC8343

MCK adjustment requires delaying MCK/$\overline{\text{MCK}}$ by (0, 1, 2 or 3)/4 clock cycles—whichever ratio guarantees the setup and hold times of clock and command/address signals at the DDR SDRAM devices. As shown in Figure 5, internal and external delays affect MCK/$\overline{\text{MCK}}$ and command/address signals.

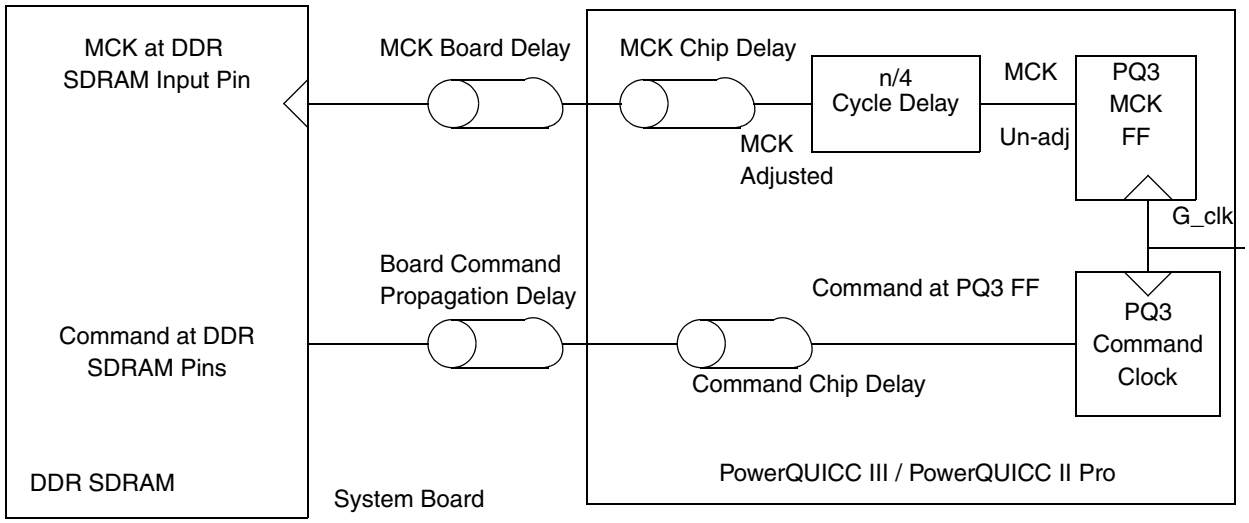


Figure 5. Conceptual MCK/DDR Command Bus Flow

Figure 6 shows the DDR SDRAM clock control register.

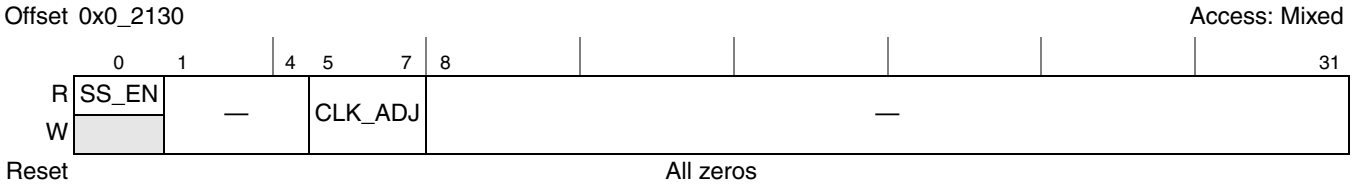


Figure 6. DDR SDRAM Clock Control Register

SS_EN should equal 1. Table 7 shows the possible CLK_ADJ values.

Table 7. CLK_ADJ Settings

CLK_ADJ	Description
000	0/4 clock cycle delay
001	1/4 clock cycle delay
010	2/4 clock cycle delay
011	3/4 clock cycle delay
100	4/4 clock cycle delay
101–111	Reserved

Table 7 shows that if CLK_ADJ = 000, MCK/$\overline{\text{MCK}}$ is launched aligned to address/command signals, as Figure 7 shows.

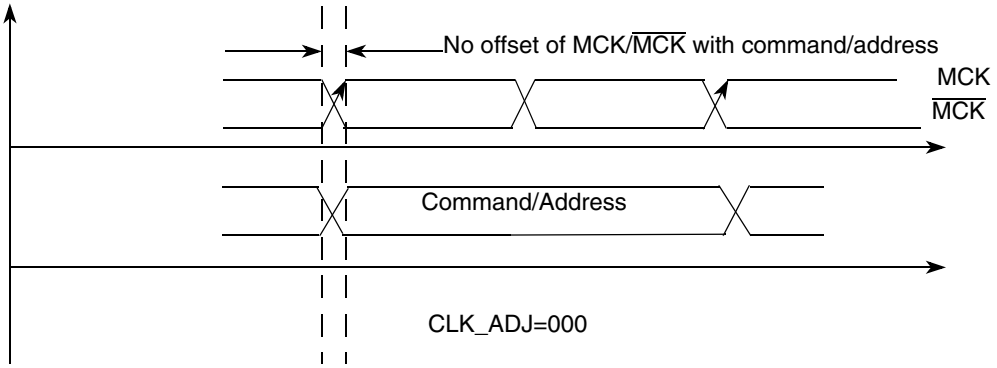


Figure 7. MCK/$\overline{\text{MCK}}$ Relative to Command/Address (CLK_ADJ = 000)

Figure 8 shows the wave forms when CLK_ADJ = 010.

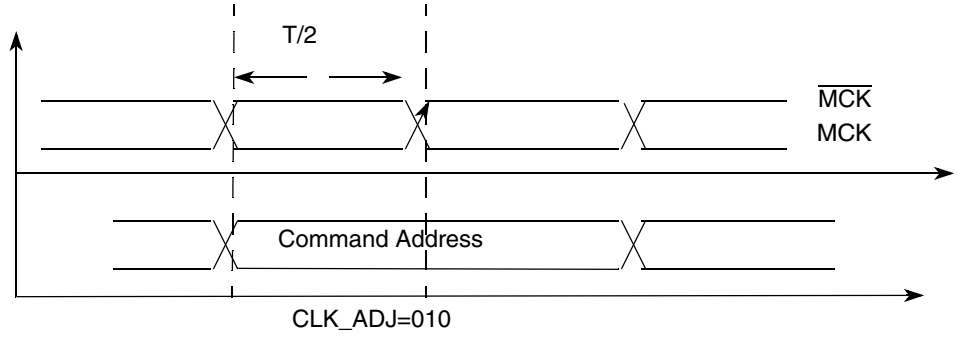


Figure 8. MCK/$\overline{\text{MCK}}$ Relative to Command/Address (CLK_ADJ = 010)

4.1.2 MCK Adjustment for MPC8540 and MPC8560

Note that in addition to this adjustment, the engineer can choose the amount of extra MCK clock board propagation delay by adding extra route length. The adjustment should be made such that the command setup and hold times are met at the DDR SDRAM devices. See application note *Hardware and Layout Design Considerations for DDR Interfaces* (AN2582) for more information on detailed timing budget calculations and, in particular, see Table 21 for itemized system delays. As stated in *MPC8540 PowerQUICC III Device Errata* (MPC8560CE) and the *MPC8560 PowerQUICC III Device Errata* (MPC8540CE), implementation of the workaround for DDR11 pre-defines the MCK/command relative phase value. This workaround overrides the PLL (self-estimated delays) and places a fixed delay based on bit field values selected in the DDRDLLCR register.

4.2 CAS to Preamble Calculation

This section describes how one can program the CPO field in the Timing_CFG_2 register. Figure 9 shows a delay model that helps us to understand the calculation method.

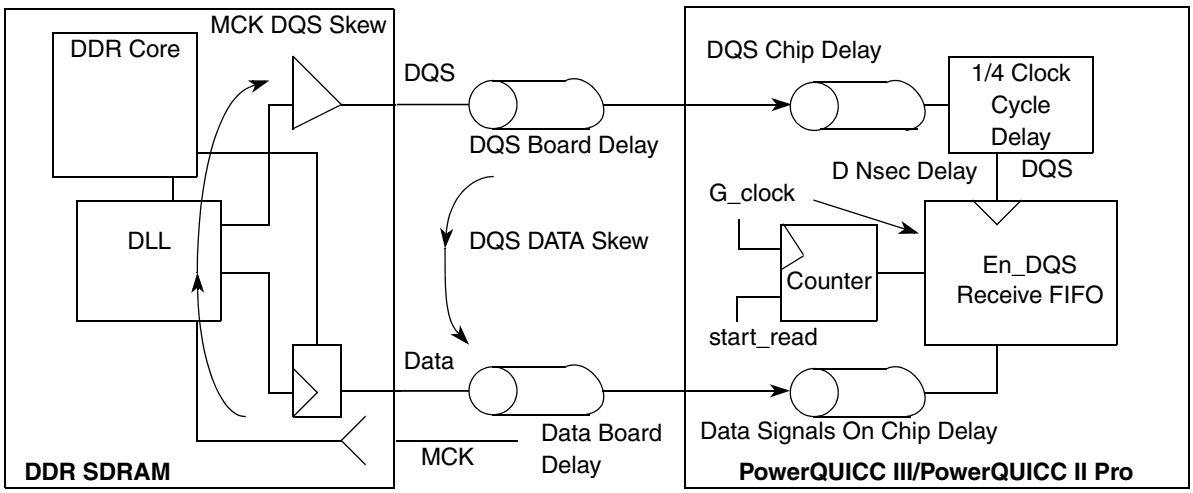


Figure 9. MCK to DQS Delay Model

The analysis estimates how many G_clock cycles are needed to let the receive FIFO look at the received DQS signal from the DDR SDRAM device in the read case. Therefore, the total amount of round trip delay from an MCK clock edge to the received DQS edge must be computed and compared with (*n* number of G_clock cycles + some delay to EN_DQS shown in Figure 9). If the round trip delay is T, G_clk cycle time is GT, and EN_DQS delay is D, then we want to know *n* such that:

```
T = n x GT + D
or
n = (T-D)/GT
```

Because T has min/max values, it is important to remember that T is the summation of all delay components in the round trip and these components have their own min and max values. Because the preamble is valid one cycle before the data strobe arrives at the controller, we must select *n* between two different *n* min/max pairs:

```
n(min-2GT)------n(min) is the T expected delays min
n(max-2GT)-----n(max) is the T expected delays max
```

We should select n to be between n(min) and n(max-2GT) to guarantee that the DDR controller always starts using a valid DQS**;** Figure 10 shows why.

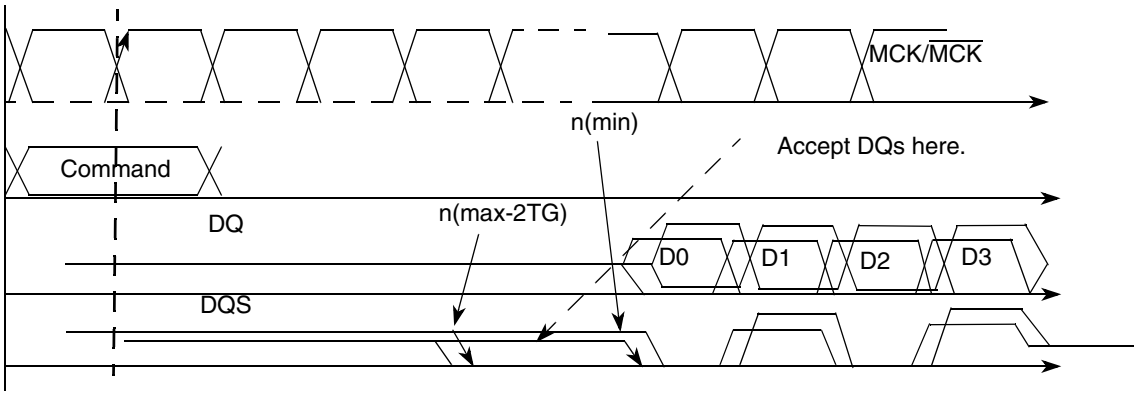


Figure 10. DDR Controller Should Always Start Using a Valid DQS

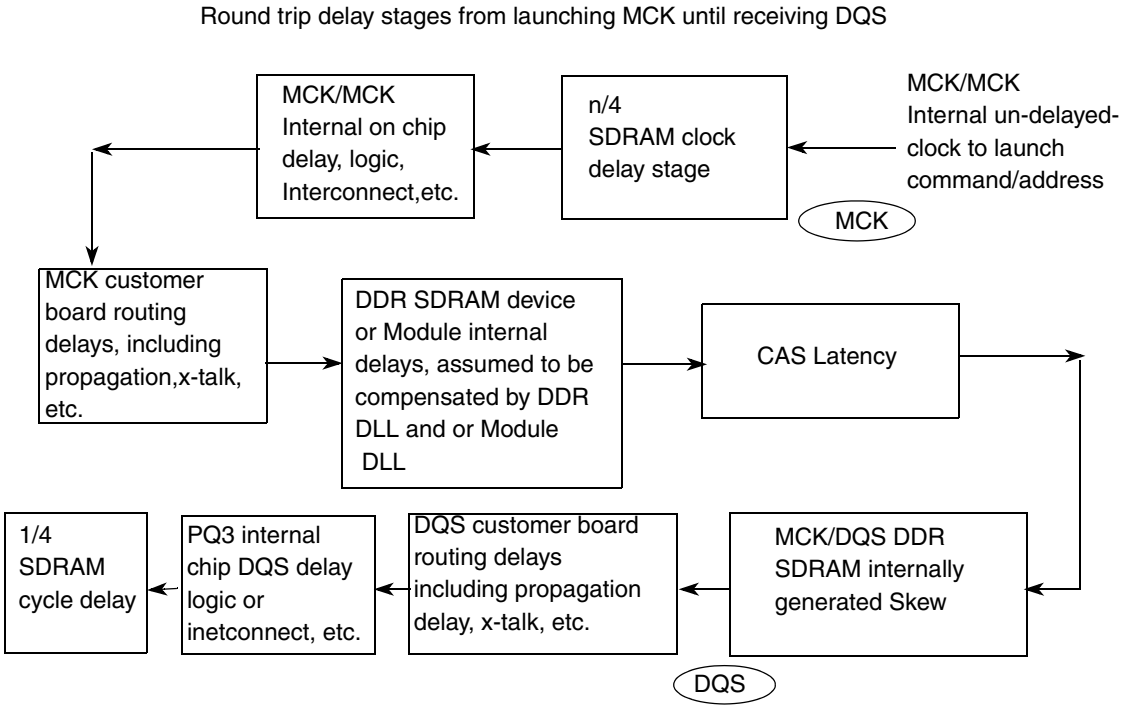


Figure 11. Round Trip Delay Stages

G_clock cycle time is half of the MCK cycle time, so the CPO field can have half the DRAM clock cycle setting granuality. The CPO parameter controls when the counter in Figure 9, which starts when a command is launched, should stop. When the counter stops, the receive FIFO expects valid data/DQS from memory. The preamble is a virtual concept indicating the earliest time when the DQS coming from memory because of read activities is valid. When we know this time span, the counter can tell the FIFO in Figure 9 to start sampling the data lines using the received DQS signals.

When we decide the optimal value of n, we should relate it to the SDRAM clock cycle count such that we consider the value of CAS latency rounded up. For example, if $\overline{\text{CAS}}$ latency is 2.5 cycles and optimal delay n is 7 G_clock cycles (that is, 3.5 SDRAM cycle), we should round up $\overline{\text{CAS}}$ latency to integer value 3 and select the CPO field that reflects [|2.5| +1/2]=[|3| + 1/2] or CPO = 0010 from Table 8. The following section describes this with a detailed example.

Table 8. Valid CPO Values

CPO	Delay	CPO	Delay
0000	CAS_LAT+1	0110	ICAS_LATI +5/2
0001	ICAS_LATI	0111	ICAS_LATI +3
0010	ICAS_LATI+ 1/2	1000	ICAS_LATI +7/2
0011	ICAS_LATI +1	1001	ICAS_LATI +4
0100	ICAS_LATI + 3/2	1010	ICAS_LATI +9/2
0101	ICAS_LATI +2	1011	ICAS_LATI +5

4.2.1 $\overline{\text{CAS}}$ to Preamble Calculation

This section describes how to set the CPO field in the TIMING_CFG_2 register for the DDR SDRAM controller. This information can be used when the workaround for DDR2 is implemented for MPC8540/MPC8560 or for source synchronous mode for other devices. The controller uses the CPO field to find out when to begin checking the data strobes during reads. The controller must begin checking the data strobe during the preamble provided by the DRAMs. To determine the proper setting for the CPO ($\overline{\text{CAS}}$-to-preamble override), you must determine two components based on the board layout (these are displayed in **boldface** throughout this document). See Table 9.

- Propagation delay of MCK($\mathbf{t_{pd\_mck(min)}}$ and $\mathbf{t_{pd\_mck(max)}}$)
- Propagation delay of MDQS[0:8] ($\mathbf{t_{pd\_mdqs(min)}}$ and $\mathbf{t_{pd\_mdqs(max)}}$)

In addition, the following parameters are needed to determine the CAS to preamble settings:

- MCK Clock Adjust (1/2 DRAM cycle if using DLL override of coarse select=1, fine select=0)
- Setting of TIMING_CFG_1[CASLAT]
- DRAM specification of MDQS/MDQ skew for given frequency
- Propagation delay of internal FIFO EN_DQS signals (D_min and D_max)

Table 9. Internal Delays and CPO Granularity for MCK and DQS Signals

Device	Total t_{dly_chip} Minimum	Total t_{dly_chip} Maximum	CPO Granularity
MPC8560, MPC8558	2.04 ns + 1/4 DRAM cycle	4.10 ns + 1/4 DRAM cycle	1/2 DRAM cycle
MPC8555E, MPC8541E	1.361 ns + 1/4 DRAM cycle	3.504 ns + 1/4 DRAM cycle	1/2 DRAM cycle
MPC8349/47/43 Rev 1.x	2.20 ns + 1/4 DRAM cycle	5.05 ns + 1/4 DRAM cycle	1/2 DRAM cycle
MPC8349/47/43 Rev 3.x	1.924 ns	4.468 ns	1/4 DRAM cycle
MPC8360/58 Controller 0 Rev 2.0	2.167 ns	4.396 ns	1/4 DRAM cycle
MPC8360 Controller 1 Rev 2.0	2.055 ns	4.181 ns	1/4 DRAM cycle
MPC8323/21 Rev 2.0	1.845 ns	4.396 ns	1/4 DRAM cycle
MPC8313	2.264 ns	5.148 ns	1/4 DRAM cycle
MPC8315/14	0.273 ns	3.519 ns	1/4 DRAM cycle
MPC8379/78/77	1.185 ns	2.701 ns	1/4 DRAM cycle
MPC8548 Rev 1.0	2.590 ns	3.868 ns	1/4 DRAM cycle
MPC8548/47/43 Rev 2.0	2.210 ns	4.171 ns	1/4 DRAM cycle
MPC8548/47/43 Rev 3.0	1.203 ns	2.465 ns	1/4 DRAM cycle
MPC8533, MPC8544	2.304 ns	3.661 ns	1/4 DRAM cycle
MPC8572 Controller 0	1.372 ns	2.914 ns	1/4 DRAM cycle
MPC8572 Controller 1	1.220 ns	2.595 ns	1/4 DRAM cycle

Table 9. Internal Delays and CPO Granularity for MCK and DQS Signals (continued)

Device	Total t_{dly_chip} Minimum	Total t_{dly_chip} Maximum	CPO Granularity
MPC8568	1.405 ns	3.567 ns	1/4 DRAM cycle
MPC8641D Controller 0 Rev 2.0	1.341 ns	2.090 ns	1/4 DRAM cycle
MPC8641D Controller 1 Rev 2.0	1.366 ns	2.017 ns	1/4 DRAM cycle
MPC8610	0.955 ns	2.288 ns	1/4 DRAM cycle
MPC8536	0.896 ns	2.474 ns	1/4 DRAM cycle
MPC8308	1.9 ns	5.8 ns	1/4 DRAM cycle
MPC8309/MPC8306S	1.8 ns	6.0 ns	1/4 DRAM cycle

4.2.2 Calculation of Total Round Trip Delay and CPO

The minimum and maximum total round trip delay (MCK launch to MDQS receive) must be calculated. The following symbols are used:

- $t_{rt\_dly(min)}$—Minimum round trip delay (from MCK launch to DQS received) remember this is related to T.
- $t_{rt\_dly(max)}$—Maximum round trip delay (from MCK launch to DQS received) remember this is related to T.
- $t_{pd\_mck(min)}$—Minimum propagation delay for MCK on board
- $t_{pd\_mck(max)}$—Maximum propagation delay for MCK on board
- $t_{pd\_mdqs(min)}$— Minimum propagation delay for MDQS on board
- $t_{pd\_mdqs(max)}$—Maximum propagation delay for MDQS on board
- $t_{dram\_skew(min)}$—Minimum MCK->MDQS skew presented by DRAM
- $t_{dram\_skew(max)}$—Maximum MCK->MDQS skew presented by DRAM
- $t_{dly\_chip(min)}$—Minimum delay for MCK/MDQS presented by chip is related to $\mathbf{t_{pd\_mck\ min}}$
- $t_{dly\_chip(max)}$—Maximum delay for MCK/MDQS presented by chip is related to $\mathbf{t_{pd\_mck\ max}}$
- t_{caslat}—CAS Latency used
- t_{clk_adjust} —Clock adjust used (1/2 DRAM cycle for DLL workaround)

NOTE: MPC8540 and MPC8560 DDR2 Workaround

For the MPC8540 and the MPC8560, the DDR2 workaround requires that DDRDLLCR[COURSE_SET] = 1 and DDRDLLCR[TAP_SEL] = 0

Following are typical settings for different parameters:

- $t_{dly\_chip(min)}$ should be derived from Table 9.
- $t_{dly\_chip(max)}$ should be derived from Table 9.
- $t_{dram\_skew}(min)$[1] =

1. These are the skews listed in the DDR1 and DDR2 JEDEC specifications.

  - — –400 ps for DDR2-667
  - — –450 ps for DDR2-533
  - — –500 ps for DDR2-400
  - — –600 ps for DDR1-333
  - — –750 ps for DDR1-266
  - — –800 ps for DDR1-200
- $t_{dram\_skew}(max)$[1] =
  - — 400 ps for DDR2-667
  - — 450 ps for DDR2-533
  - — 500 ps for DDR2-400
  - — 600 ps for DDR1-333
  - — 750 ps for DDR1-266
  - — 800 ps for DDR1-200
- t_{caslat} = 4 or 5 DRAM cycles for DDR2-667; 3 or 4 DRAM cycles for DDR2-533; 3 or 4 DRAM cycles for DDR2-400; 2.5 DRAM cycles for DDR1-333; 2 DRAM cycles for DDR1-266, DDR1-200
  - — 12000 ps or 15000 ps for DDR2-667 (derived from [4 * 3000 ps] or [5 * 3000 ps])
  - — 11250 ps or 15000 ps for DDR2-533 (derived from [3 * 3750 ps] or [4 * 3750 ps])
  - — 15000 ps or 20000 ps for DDR2-400 (derived from [3 * 5000 ps] or [4 * 5000 ps])
  - — 15000 ps for DDR1-333 (derived from 2.5 * 6000 ps)
  - — 15000 ps for DDR1-266 (derived from 2 * 7500 ps)
  - — 20000 ps for DDR1-200 (derived from 2 * 10000 ps)
- t_{clk_adjust} = 1/2 DRAM cycle
  - — 1500 ps for DDR2-667
  - — 1875 ps for DDR2-533
  - — 2500 ps for DDR2-400
  - — 3000 ps for DDR1-333
  - — 3750 ps for DDR1-266
  - — 5000 ps for DDR1-200

The round trip minimum and maximum delays are calculated as:

- $t_{rt\_dly(min)} = t_{dly\_chip(min)} + t_{dram\_skew(min)} + t_{caslat} + t_{clk\_adjust} + \mathbf{t_{pd\_mck(min)}} + \mathbf{t_{pd\_mdqs(min)}}$
- $t_{rt\_dly(max)} = t_{dly\_chip(max)} + t_{dram\_skew(max)} + t_{caslat} + t_{clk\_adjust} + \mathbf{t_{pd\_mck(max)}} + \mathbf{t_{pd\_mdqs(max)}}$

From these calculations, the TIMING_CFG_2[CPO] field can be programmed to meet the minimum and maximum requirements of the data strobe return. The equation above gives the round trip time until the first rising edge of the receive data strobe. It is important to calculate the CPO field based on the fact that the preamble will be driven low for almost a full cycle before the return data strobe.

1. These are the skews listed in the DDR1 and DDR2 JEDEC specifications.

4.2.3 Example Calculation (DDR1)

As an example calculation, consider a DDR1-333 (CASLAT=2.5) system, for MPC8560 device.

- $t_{rt\_dly(min)} = t_{dly\_chip(min)} + t_{dram\_skew(min)} + t_{caslat} + t_{clk\_adjust} + t_{pd\_mck(min)} + t_{pd\_mdqs(min)}$
- $t_{rt\_dly(min)} = (2040 \text{ ps} + 1/4 \text{ cycle}) + (-600 \text{ ps}) + (15000 \text{ ps}) + (3000 \text{ ps}) + t_{pd\_mck(min)} + t_{pd\_mdqs(min)}$
- $t_{rt\_dly(min)} = (3540 \text{ ps}) + (-600 \text{ ps}) + (15000 \text{ ps}) + (3000 \text{ ps}) + t_{pd\_mck(min)} + t_{pd\_mdqs(min)}$
- $t_{rt\_dly(min)} = 20940 \text{ ps} + t_{pd\_mck(min)} + t_{pd\_mdqs(min)}$
- $t_{rt\_dly(max)} = t_{dly\_chip(max)} + t_{dram\_skew(max)} + t_{caslat} + t_{clk\_adjust} + t_{pd\_mck(max)} + t_{pd\_mdqs(max)}$
- $t_{rt\_dly(max)} = (4100 \text{ ps} + 1/4 \text{ cycle}) + (600 \text{ ps}) + (15000 \text{ ps}) + (3000 \text{ ps}) + t_{pd\_mck(min)} + t_{pd\_mdqs(min)}$
- $t_{rt\_dly(max)} = (5600 \text{ ps}) + (600 \text{ ps}) + (15000 \text{ ps}) + (3000 \text{ ps}) + t_{pd\_mck(min)} + t_{pd\_mdqs(min)}$
- $t_{rt\_dly(max)} = 24200 \text{ ps} + t_{pd\_mck(min)} + t_{pd\_mdqs(min)}$

For example, assume the following propagation delays for the board:

- $t_{pd\_mck(min)} = 800$ ps
- $t_{pd\_mdqs(min)} = 800$ ps
- $t_{pd\_mck(max)} = 1000$ ps
- $t_{pd\_mdqs(max)} = 1000$ ps
- $t_{rt\_dly(min)} = 20940 \text{ ps} + 800 \text{ ps} + 800 \text{ ps}$
- $t_{rt\_dly(max)} = 24200 \text{ ps} + 1000 \text{ ps} + 1000 \text{ ps}$
- $t_{rt\_dly(min)} = 22540$ ps
- $t_{rt\_dly(max)} = 26200$ ps

The preamble driven by the DRAM is guaranteed to be between 90 percent and 110 percent of a DRAM cycle. To obtain the worst case scenario, 90 percent of a preamble would be used when finding the valid times for the data strobes. However, for the maximum round trip case, the DRAM parameter, t_{lz} takes priority over the 90 percent preamble minimum. This t_{lz} parameter defines the time from an MCK edge until the preamble is driven active. Depending upon the frequency, the preamble for the maximum round trip scenario may be slightly less than a full DRAM cycle. The following provides the minimum preamble for the minimum and maximum round trip scenarios.

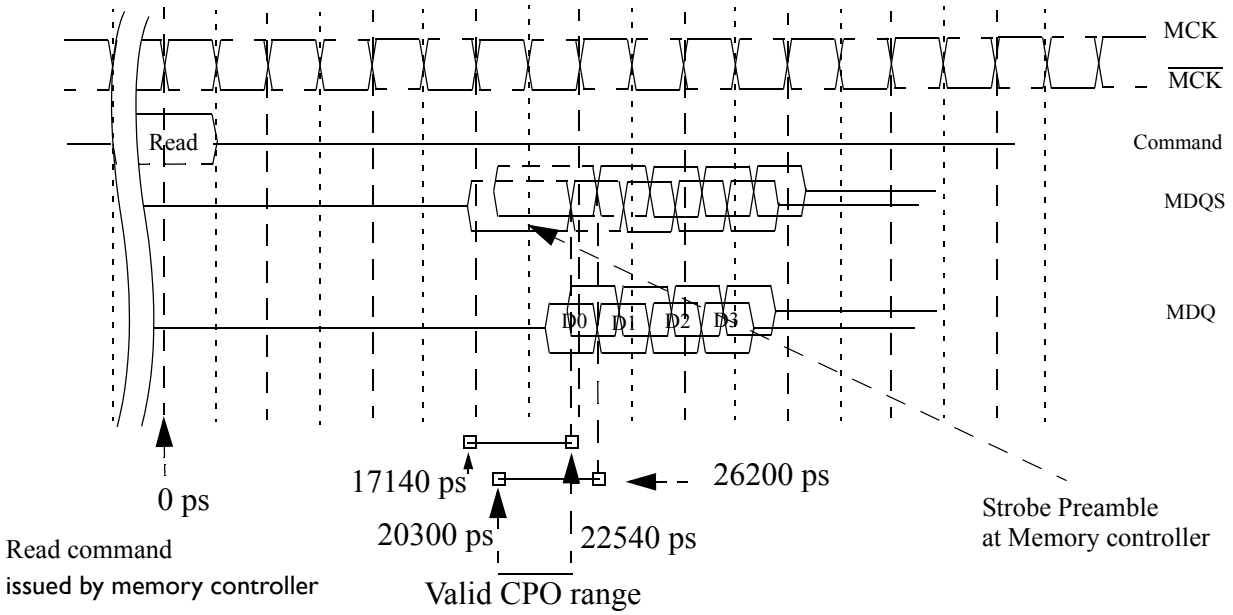
- For the minimum requirements, the minimum preamble is 90 percent of a DRAM cycle.
  - DDR2-667: 2700 ps
  - DDR2-533: 3375 ps
  - DDR2-400: 4500 ps
  - DDR1-333: 5400 ps
  - DDR1-266: 6750 ps
  - DDR1-200: 9000 ps
- For the maximum requirements, the minimum preamble is [1 DRAM cycle - ($t_{lz(max)}$ - t_{dram_skew}).
  - DDR2-667: $t_{lz(max)}$=450ps, minimum preamble = [3000 ps - (450 ps - 400 ps)]= 2950 ps
  - DDR2-533: $t_{lz(max)}$=500ps, minimum preamble = [3750 ps - (500 ps - 450 ps)]= 3700 ps
  - DDR2-400: $t_{lz(max)}$=600ps, minimum preamble = [5000 ps - (600 ps - 500 ps)]= 2900 ps

— DDR1-333: $t_{lz(max)}$=700ps, minimum preamble = [6000 ps - (700 ps - 600 ps)]= 5900 ps
— DDR1-266: $t_{lz(max)}$=750ps, minimum preamble = [7500 ps - (750 ps - 750 ps)]= 7500ps
— DDR1-200: $t_{lz(max)}$=800ps, minimum preamble = [10000 ps - (800 ps - 800 ps)]= 10000 ps

Therefore, the valid times to look for the strobes in the example of DDR1-333 are:

- Minimum Requirements: 17140 ps to 22540 ps which is derived from:
  — [$t_{rt\_dly(min)}$ - (90%) * 1 DRAM cycle] to $t_{rt\_dly(min)}$
  — [22540 ps - 0.9* 6000 ps] to 22540 ps
  — [22540 ps - 5400 ps] to 22540 ps
  — 17140 ps to 22540 ps
- Maximum Requirements: 20300 ps to 26200 ps which is derived from:
  — [$t_{rt\_dly(max)}$ - (1 DRAM cycle - ($t_{lz(max)}$ - t_{dram_skew}))] to $t_{rt\_dly(max)}$)
  — [26200 ps - (6000 ps - (700 ps -600 ps))] to 26200 ps
  — [26200 ps - 5900 ps] to 26200 ps
  — 20300 ps to 26200 ps

The figure below shows the graphical representation of the above calculation.



The TIMING_CFG_2[CPO] field must be programmed so that a valid time is chosen that meets the minimum and maximum requirements. Therefore, a value between 20300 ps and 22540 ps must be chosen.

— [$t_{rt\_dly(max)}$ - (1 DRAM cycle - ($t_{lz(max)}$ - t_{dram_skew}))] < CPO < $t_{rt\_dly(min)}$
— 20300 ps < CPO < 22540 ps

For this example, 1/2 applied cycle granularity is assumed to be provided by the CPO field (derived from Table 9). For DDR1-333, this allows 3000 ps granularity. In this example, a delay of 21000 ps would satisfy this requirement, which is equal to 3.5 DRAM cycles.

— 20300 ps < CPO < 22540 ps

— 20300 ps < 21000 ps < 22540 ps
— CPO = 21000 ps = 3.5 DRAM cycles = 2.5 + 1 DRAM cycles
— CPO = CASLAT + 1
— TIMING_CFG_2[CPO] = 0000

The default CASLAT + 1 cycle would be the correct setting for the TIMING_CFG_2[CPO] setting. In addition, a CPO value of 0010 would yield the same result.

— CPO = 21000 ps = 3.5 DRAM cycles = |2.5 (round up to next integer)| + 1/2
— CPO = |CASLAT| + 1/2
— TIMING_CFG_2[CPO] = 0010

This value is decoded to |CASLAT| + 1/2, where |CASLAT| is the CASLAT rounded up to the next integer. For a CASLAT of 2.5 DRAM cycles, |CASLAT| + 1/2 is equal to 3.5 DRAM cycles. Note that all decagons other than the default for the CPO field round CASLAT up to the next integer value before adding an offset.

4.2.4 Example Calculation (DDR2)

In this section the CPO calculation is done for a DDR2 example.

consider a DDR2-533 system for MPC8548 revision 2.x silicon with following memory controller and board assumption:

- Operating in memory bus frequency of 200 Mhz or DRAM cycle of 5000 ps
- t_{clk_adjust} = 1/2 clock = 5000 ps * 1/2 = 2500 ps
- $t_{pd\_mck(min)}$ = 800 ps
- $t_{pd\_mdqs(min)}$ = 800 ps
- $t_{pd\_mck(max)}$ = 1000 ps
- $t_{pd\_mdqs(max)}$ = 1000 ps

Here we work with the formulas to define the valid CPO range:

First the min and max total time delays:

- $t_{rt\_dly(min)} = t_{dly\_chip(min)} + t_{dram\_skew(min)} + t_{Readlat} + t_{clk\_adjust} + t_{pd\_mck(min)} + t_{pd\_mdqs(min)}$ as explained in Section 4.2.2, “Calculation of Total Round Trip Delay and CPO”
- $t_{rt\_dly(min)}$ = 2210 ps + (– 500 ps) + $t_{Readlat}$ + 2500 ps+ 800 ps + 800 ps
- $t_{rt\_dly(min)}$ = 5810 ps + $t_{Readlat}$
- $t_{rt\_dly(max)} = t_{dly\_chip(max)} + t_{dram\_skew(max)} + t_{Readlat} + t_{clk\_adjust} + t_{pd\_mck(max)} + t_{pd\_mdqs(max)}$ as explained in Section 4.2.2, “Calculation of Total Round Trip Delay and CPO”
- $t_{rt\_dly(max)}$ = 4171 ps + 500 ps + $t_{Readlat}$ + 2500 ps+ 1000 ps+ 1000 ps
- $t_{rt\_dly(max)}$ = 9171 ps + $t_{Readlat}$

Next as explained in Section 4.2.3, “Example Calculation (DDR1),” the CPO value range is obtained from:

- $[t_{rt\_dly(max)} – (1 \text{ DRAM cycle} - (t_{lz(max)} – t_{dram\_skew}))] < CPO < t_{rt\_dly(min)}$

- [9171 ps + $t_{Readlat}$- (5000 ps- (600 ps - 500 ps))] < CPO < [5810 ps + $t_{Readlat}$]. Note that t_{lz} parameter is frequency dependent and hence its value should be selected based on the frequency of operation rather than the SDRAM part capability.
- [9171 ps + $t_{Readlat}$ – 4900 ps] < CPO < [5810 ps + $t_{Readlat}$]
- [4271 ps + $t_{Readlat}$] < CPO < [5810 ps + $t_{Readlat}$]

Rewriting the above formula in terms of number of cycles instead of pico second:

- [(4271 ps/5000 ps) + $t_{Readlat}$] < CPO < [(5810 ps/5000 ps)+ $t_{Readlat}$]
- 0.8542 + $t_{Readlat}$ < CPO < 1.162 + $t_{Readlat}$
- $t_{Readlat}$ + 0.8542 < CPO < $t_{Readlat}$ + 1.162

We can only select CPO = $t_{Readlat}$ + X (in 1/4 clock cycle granuality for MPC 8548), so for this example, the value of CPO = $t_{Readlat}$ + 1 and the corresponding register setting of TIMING_CFG_2[CPO] = 00110 would be the correct setting. In this example, only one valid CPO is possible. However, if more than one CPO value is possible in the valid CPO range, the one with the higher margin should be selected.

It can be observed from the above calculation that the CPO value is independent of the value of Read_LAT = CASLAT + Add_LAT. This means, when the value of CAS_LAT or Add_LAT in memory controller are changed, the CPO value is not affected and shall remain the same. Please note that the independence of the CPO value from the Read_LAT is only true when CAS_LAT value is an integer value which is true for all DDR2. For DDR1 please follow the calculation steps provided in previouse section, because CASLAT can have non-integer values.

5 Documentation References

Consult the reference documentation in Table 10 for additional information.

Table 10. Documentation References

Document Title	Document ID
MPC85xx Hardware Specifications	MPC85*xx*EC
MPC83xx Hardware Specifications	MPC83*xx*EC
MPC85xx PowerQUICC III Integrated Communications Processor Reference Manual	MPC85*xx*RM
MPC83xx PowerQUICC III Integrated Communications Processor Reference Manual	MPC83*xx*RM
Hardware and Layout Design Considerations for DDR Interfaces	AN2582
Hardware and Layout Design Considerations for DDR2 Interfaces	AN2910
PowerQUICC DDR2 SDRAM Controller Register Setting Considerations	AN3369

6 Document Revision History

Table 11 provides a revision history for this document.

Table 11. Document Revision History

Rev. Number	Date	Substantive Change(s)
11	07/2014	In Table 9, “Internal Delays and CPO Granularity for MCK and DQS Signals,” added rows for MPC8548/47/43 Rev 3.0
10	02/2014	In section, Section 4.1.2, “MCK Adjustment for MPC8540 and MPC8560,” removed table.
9	02/2011	In Table 9, “Internal Delays and CPO Granularity for MCK and DQS Signals,” added rows for MPC8308, MPC8309, and MPC8306S.
8	12/2008	Added a row at the end of Table 9 for the MPC8536 device info.
7	3/2008	Modified first paragraph in Section 4, “Source Synchronous Mode” and updated title of Figure 3 accordingly.
6	10/2007	Added Section 4.2.4, “Example Calculation (DDR2) and added to Section 4.2.3, “Example Calculation (DDR1) added to Table 10
5	08/2007	Added and modified information in Table 9
4	03/2007	Added information to table 9
3	08/2006	Added sections 4.2.1
2	11/2004	Added Section 4, “Source Synchronous Mode”
1	11/2004	Various formatting changes
0	12/2003	Initial release of document.

How to Reach Us:

Home Page:
freescale.com

Web Support:
freescale.com/support

Document Number: AN2583
Rev. 11
07/2014