DDR PHY Interface (DFI) Specification

Version 2.1
22 June 2009

DENALI SOFTWARE, INC.
1000 Hamlin Court
Sunnyvale, CA 94089
Tel: (408) 743-4200
Fax: (408) 743-4209

info@denali.com
sales@denali.com
www.denali.com/support
www.ememory.com

Denali Software, Inc. Palo Alto, CA 94303

Release Information

Rev #	Date	Change
1.0	30 Jan 2007	Initial Release
2.0	17 Jul 2007	Modifications/Additions for DDR3 Support
2.0	21 Nov 2007	Additional modifications/additions for DDR3 support. Added read and write leveling. Changes approved by the Technical Committee for DDR3 support.
2.0	21 Dec 2007	Removed references to data eye training for PHY Evaluation mode, added a gate training-specific mode signal, corrected references and clarified read leveling.
2.0	11 Jan 2008	Modified wording; standardized notations in figures, clarified terminology for read and write leveling.
2.0	26 Mar 2008	Added timing parameter t_{rdlvl_en} and t_{wrlvl_en}, signal dfi_rdlvl_edge
2.1	2 Oct 2008	Added initial LPDDR2 support and corrected minor errors from 2.0 release
2.1	24 Nov 2008	Added frequency change protocol, signal timing definitions, t_{rdlvl_load} and t_{wrlvl_load} timing parameters and adjusted diagrams accordingly
2.1	30 Jan 2009	Added DFI logo
2.1	31 Mar 2009	Updated width of dfi_rdlvl_edge, corrected erroneous figures, updated t_{rdlvl_en} and t_{wrlvl_en} definitions
2.1	20 May 2009	Added low power control interface, modified leveling request signal description to include frequency change, added dfi_data_byte_disable signal and t_{phy_wrdata} timing parameters. Added DIMM support to the status interface and updated frequency ratios from an example to a defined method. Updated frequency ratios information for new proposals, modified default values and requirements for some training interface signals, incorporated LPDDR2 training operations changes
2.1	22 June 2009	Expanded frequency ratio information to include vectored read data, expanded use of dfi_init_start, added timing diagrams for 1:4 frequency ratio systems

Proprietary Notice

No part of this document may be copied or reproduced in any form or by any means without prior written consent of Denali.

Denali makes no warranties with respect to this documentation and disclaims any implied warranties of merchantability or fitness for a particular purpose. Information in this document is subject to change without notice. Denali assumes no responsibility for any errors that may appear in this document.

Except as may be explicitly set forth in such agreement, Denali does not make, and expressly disclaims, any representations or warranties as to the completeness, accuracy, or usefulness of the information contained in this document. Denali does not warrant that use of such information will not infringe any third party rights, nor does Denali assume any liability for damages or costs of any kind that may result from use of such information.

RESTRICTED RIGHTS LEGEND

Use, duplication, or disclosure by the Government is subject to restrictions as set forth in subparagraphs (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013.

Destination Control Statement

All technical data contained in this product is subject to the export control laws of the United States of America. Disclosure to nationals of other countries contrary to United States law is prohibited. It is the reader's responsibility to determine the applicable regulations and to comply with them.

Trademarks

Denali and the Denali logo are registered trademarks of Denali Software, Inc.

All other products or brand names mentioned are trademarks or registered trademarks of their respective holders.

End User License Agreement

1. Subject to the provisions of Clauses 2, 3, 4, 5 and 6, Denali hereby grants to licensee ("Licensee") a perpetual, non-exclusive, nontransferable, royalty free, worldwide copyright license to use and copy the DFI (DDR PHY Interface) specification (the "DFI Specification") for the purpose of developing, having developed, manufacturing, having manufactured, offering to sell, selling, supplying or otherwise distributing products which comply with the DFI Specification.
2. THE DFI SPECIFICATION IS PROVIDED "AS IS" WITH NO WARRANTIES EXPRESS, IMPLIED OR STATUTORY, INCLUDING BUT NOT LIMITED TO ANY WARRANTY OF SATISFACTORY QUALITY, MERCHANTABILITY, NONINFRINGEMENT OR FITNESS FOR A PARTICULAR PURPOSE.
3. No license, express, implied or otherwise, is granted to Licensee, under the provisions of Clause 1, to use Denali's or any other person or entity participating in the development of the DFI Specification listed herein (individually "Participant," collectively "Participants") trade name, or trademarks in connection with the DFI Specification or any products based thereon. Nothing in Clause 1 shall be construed as authority for Licensee to make any representations on behalf of Denali or the other Participants in respect of the DFI Specification.
4. NOTWITHSTANDING ANYTHING ELSE WILL DENALI'S TOTAL AGGREGATE LIABILITY FOR ANY CLAIM, SUIT, PROCEEDING OR OTHERWISE, RELATING IN ANYWAY TO THE DFI SPECIFICATION EXCEED $1.00USD.
5. NOTWITHSTANDING ANYTHING ELSE WILL ANY PARTICIPANT'S TOTAL AGGREGATE LIABILITY FOR ANY CLAIM, SUIT, PROCEEDING OR OTHERWISE, RELATING IN ANYWAY TO THE DFI SPECIFICATION EXCEED $1.00USD.

6. Licensee agrees that Denali and the Participants may use, copy, modify, reproduce and distribute any written comments or suggestions (“Communications”) provided regarding the DFI Specification by Licensee and that Licensee will not claim any proprietary rights in the DFI Specification, or implementations thereof by any Participant or third party, as a result of the use of the Communications in developing or changing the DFI Specification. Denali and the participants will have no confidentiality obligations with respect to the Communications and Licensee will not include any confidential information of Licensee or any third party in any Communications.

Participants

ARM Denali Intel LSI Samsung ST

CONTENTS

1.0 Overview

The DDR PHY Interface (DFI) is an interface protocol that defines the connectivity between a DDR memory controller (MC) and a DDR physical interface (PHY) for DDR1, LPDDR1, DDR2, LPDDR2 and DDR3 memory devices. The protocol defines the signals, signal relationships, and timing parameters required to transfer control information and data to and from the DRAM devices over the DFI. This interface does not encompass all of the features of the MC or the PHY, nor does it put any restrictions on how the PHY or the MC interface to other aspects of the system such as DFT, other system calibration capabilities, or other signals that may exist between the MC and the PHY for a particular implementation.

The widths of DFI signals are dependent on the system configuration. A glossary of terms used in this specification can be found in Section 6.0, “Glossary”.

Changes in the DFI protocol between version 1.0, version 2.0 and version 2.1 may result in incompatibilities between MCs and PHYs designed to adhere to different versions of the standard. MCs and PHYs designed to version 2.0 may not be backwards compatible. Changes in the DFI protocol between version 2.0 and version 2.1 will maintain backward compatibility; however, some features supported by a DFI 2.1 device may not be supported by a DFI 2.0 device. Refer to the DFI 2.1 Errata for more information on differences between the two versions of the specification. The low power, frequency ratio and frequency change protocols added in DFI 2.1 are optional features and are not required for DFI compatibility.

2.0 Architecture

The DDR PHY Interface specification does not specify timing values for signaling between the MC and the PHY. The only requirement is that the DFI clock must exist, and all signals defined by the DFI are required to be driven by registers referenced to a rising edge of the DFI clock. There are no restrictions on how these signals are received, nor are there rules dictating the source of the DFI clock. Compatibility between the MC and the PHY at given frequencies is relative based on the specification of both the output timing for signals driven and the setup and hold requirements for receiving these signals on the DFI. Refer to the Section 5.0, “Signal Timing” for more information on timing.

The DFI specification includes signal and timing parameter descriptions required for DFI compliance. DFI compatibility is dependent on the widths and values of signals and timing parameters provided by the MC and the PHY. Fully compliant DFI devices may be incompatible if their DFI signal widths and/or their timing parameters are inconsistent, i.e., they may or may not be able to communicate via the DFI if their system settings are inconsistent or their timing parameters are out-of-range.

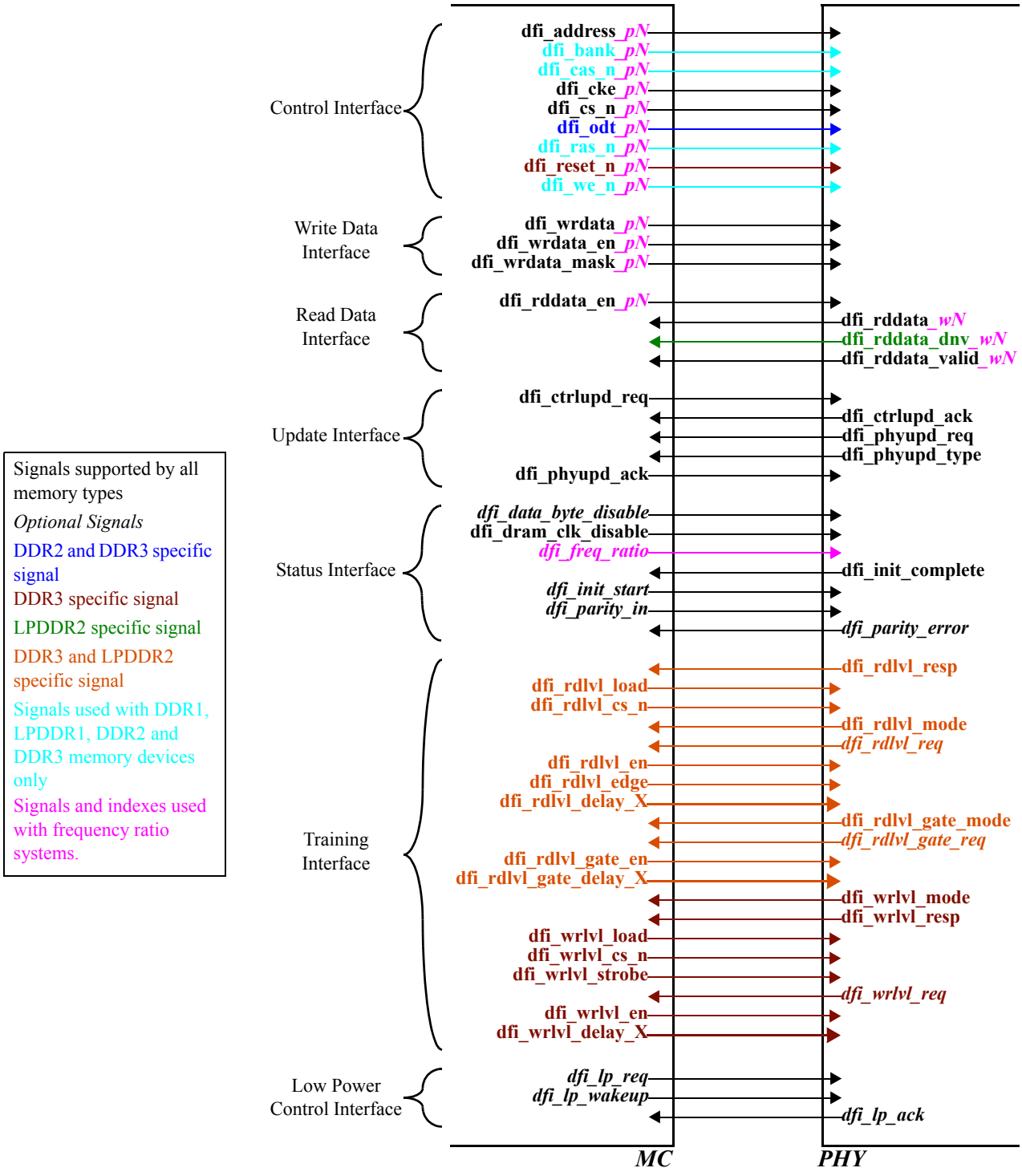
The DFI does not dictate absolute latencies for control signals, read data or write data to or from the DRAM devices. However, the DFI does include timing parameter definitions that must be specified by the MC, the PHY, or the system as a whole for DFI compliance. These timing parameters define signal timing relationships for the DFI protocol to send control, read and write data across the DFI. The values supported for the various timing parameters are defined by the MC and the PHY individually. Compatibility between the MC and the PHY depends on the values and ranges of these timing parameters supported by each component individually. The DFI specification does not dictate a fixed range of values that must be supported by each device.

The DFI specification allows certain timing parameters to be specified as fixed values, maximum values or as constants based on other values in the system. These timing parameters must be held constant while commands are being executed on the DFI bus; however, if necessary, these values may be changed while the bus is idle.

The DFI specification supports an MC and PHY operating in either matched frequency or a frequency ratio, or both. In a frequency ratio system, the MC and the PHY are not operating at the same frequency. These systems operate at a frequency ratio of 1:2 or 1:4, depending on the relationship of the reference clocks for the MC and the PHY. For frequency ratio systems, the control interface, write data interface and the read data enable signals will be replaced with phase-specific signals with a suffix of “_pN” with the phase number N (ex: **dfi_wrdata_pN**) which allows the MC to drive multiple commands in a single clock cycle. The other signals of the read data interface will be replaced with DFI data word-specific signals with a suffix of “_wN” with the DFI data word N (ex: **dfi_rddata_en**) to distinguish how memory words are transferred across the DFI bus. For all signal types, the suffix for phase 0/word 0 is optional. Frequency ratio support is optional for both the MC and the PHY. Refer to Section 4.7, “Frequency Ratios Across the DFI” for more information on this feature. The MC and the PHY must operate at a common frequency ratio (1:1, 1:2 or 1:4) to be DFI-interoperable.

In addition, the DFI specification includes optional protocols for handling low power and system frequency change. Support for these protocols are not required for DFI compliance. Refer to Section 4.8, “Frequency Changing” and Section 4.11, “Low Power Control Handshaking” for more information on these protocols.

FIGURE 1. *Block Diagram*



3.0 Interface Signal Groups

The DFI is subdivided into the following interface groups:

- Control Interface
- Write Data Interface
- Read Data Interface
- Update Interface
- Status Interface
- Training Interface
- Low Power Control Interface

The control interface is a reflection of the DRAM control interface including address, bank, chip select, row strobe, column strobe, write enable, clock enable and ODT control, as applicable for the memory technology. The write data and read data interfaces are used to send valid write and receive valid read data across the DFI. The update interface provides an ability for the PHY or the MC to interrupt and stall the DFI. The status interface is used for system initialization and feature support, as well as to control the presence of valid clocks to the DRAM interface. The training interface is used for executing data eye training, gate training and write leveling operations. The low power control interface is used to control low power modes for the PHY.

3.1 Control Interface

The DFI specification includes signals required to drive the memory address, command, and control signals to the DRAM devices. These signals are intended to be passed to the DRAM devices in a manner that maintains the timing relationship of these signals on the DFI. The actual delay introduced between the DFI interface and the DRAM interface is defined by the $\mathbf{t_{ctrl\_delay}}$ timing parameter.

Some signals of the control interface are memory technology-specific and are only required if the interface is being used for the associated technology. The signal **dfi_reset_n** is specific to DDR3 memory systems and the **dfi_odt** signal is specific to DDR2 and DDR3 memory systems. The signal **dfi_rddata_dnv** is specific to LPDDR2 memory systems.

For frequency ratio systems, the signals of the control interface will be replicated into phase-specific signals with a suffix of "_pN" that defines the signal value for each phase N of the DFI PHY clock. Phase 0 may exclude the suffix if desired. The MC may issue commands on any phases to communicate with the PHY. For example, the MC may always issue commands only on phase 0, or may issue commands on any combination of phases. The PHY must be able to accept a command on any and all phases for DFI frequency ratio compliance.

For LPDDR2 memory systems, the CA bus will be mapped to the **dfi_address** bus. While several mapping schemes exist, a single mapping is required for interoperability between DFI 2.1 MCs and PHYs. The implementation will solely use the **dfi_address** bus and require that the **dfi_bank**, **dfi_ras_n**, **dfi_cas_n** and **dfi_we_n** signals must be held at constant values. The **dfi_address** bus must have a minimum of 20 bits to hold the LPDDR2 rising and falling DDR Command/Address (CA) bus for the entire clock period. The PHY is responsible for selecting between the rising and falling CA phases and sending a double data rate, 10-bit output to the LPDDR2 memory. The LPDDR2 interface mapping is detailed in Table 1, “Bit Definitions of the dfi_address bus for LPDDR2”.

TABLE 1. *Bit Definitions of the **dfi_address** bus for LPDDR2*

dfi_address		19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CA Bus												LPDDR2 Address1									
												9	8	7	6	5	4	3	2	1	0
		LPDDR2 Address2																			
		9	8	7	6	5	4	3	2	1	0										

More information on the control interface is provided in Section 4.2, “Control Signals”. The signals and parameter in the control interface are listed in Table 2 and Table 3.

TABLE 2. *Control Signals*

Signal	From	Width	Default	Description
dfi_address or **dfi_address_pN**	MC	DFI Address Width	- [a]	DFI address bus. These signals define the address information that is intended for the DRAM memory devices for all control commands. The PHY must preserve the bit ordering of the **dfi_address** signals when reflecting this data to the DRAM devices. For frequency ratio systems, the address signals will be replicated into phase-specific **dfi_address_pN** signals that define the address for each phase N of the PHY clock. For LPDDR2 memory systems, the **dfi_address** bus maps to the CA bus as described in Section 3.1, “Control Interface”.
dfi_bank or **dfi_bank_pN**	MC	DFI Bank Width	- [b]	DFI bank bus. These signals define the bank information that is intended for the DRAM devices for all control commands. The PHY must preserve the bit ordering of the **dfi_bank** signals when reflecting this data to the DRAM devices. For frequency ratio systems, the bank signals will be replicated into phase-specific **dfi_bank_pN** signals that define the bank for each phase N of the PHY clock. These signals are only applicable for DDR1, LPDDR1, DDR2 and DDR3 memory systems. For LPDDR2 memory systems, these signals must be held in the idle state.

TABLE 2. *Control Signals*

Signal	From	Width	Default	Description
dfi_cas_n or **dfi_cas_n_pN**	MC	DFI Control Width	0x1	DFI column address strobe bus. These signals define the CAS information that is intended for the DRAM devices for all control commands. For frequency ratio systems, the CAS signals will be replicated into phase-specific **dfi_cas_n_pN** signals that define the column for each phase N of the PHY clock. These signals are only applicable for DDR1, LPDDR1, DDR2 and DDR3 memory systems. For LPDDR2 memory systems, these signals must be held in the idle state.
dfi_cke or **dfi_cke_pN**	MC	DFI Chip Select Width	0x0 [c] 0x1 [c]	DFI clock enable bus. These signals define the CKE information that is intended for the DRAM devices for all control commands. For frequency ratio systems, the CKE signals will be replicated into phase-specific **dfi_cke_pN** signals that define the clock enables for each phase N of the PHY clock. The MC must drive these signals in all phases. The PHY must be able to accept a command on any and all phases for DFI frequency ratio compliance.
dfi_cs_n or **dfi_cs_n_pN**	MC	DFI Chip Select Width	0x1	DFI chip select bus. These signals define the chip select information that is intended for the DRAM devices for all control commands. For frequency ratio systems, the CS signals will be replicated into phase-specific **dfi_cs_n_pN** signals that define the chip select for each phase N of the PHY clock.
dfi_odt or **dfi_odt_pN**	MC	DFI Chip Select Width	0x0	DFI on-die termination control bus. These signals define the ODT information that is intended for the DRAM devices for all control commands. These signals are only required for DFI DDR2 and DDR3 support. For frequency ratio systems, the ODT signals will be replicated into phase-specific **dfi_odt_pN** signals that define the termination for each phase N of the PHY clock. The MC must drive these signals in all phases. The PHY must be able to accept a command on any and all phases for DFI frequency ratio compliance.
dfi_ras_n or **dfi_ras_n_pN**	MC	DFI Control Width	0x1	DFI row address strobe bus. These signals define the RAS information that is intended for the DRAM devices for all control commands. For frequency ratio systems, the RAS signals will be replicated into phase-specific **dfi_ras_n_pN** signals that define the row for each phase N of the PHY clock. These signals are only applicable for DDR1, LPDDR1, DDR2 and DDR3 memory systems. For LPDDR2 memory systems, these signals must be held in the idle state.

TABLE 2. *Control Signals*

Signal	From	Width	Default	Description
dfi_reset_n or **dfi_reset_n_pN**	MC	DFI Chip Select Width	0x0	DFI reset bus. These signals define the RESET information that is intended for the DRAM memory devices for all control commands. These signals are only required for DFI DDR3 support. For frequency ratio systems, the reset signals will be replicated into phase-specific **dfi_reset_n_pN** signals that define the reset for each phase N of the PHY clock.
dfi_we_n or **dfi_we_n_pN**	MC	DFI Control Width	0x1	DFI write enable bus. These signals define the WEN information that is intended for the DRAM devices for all control commands. For frequency ratio systems, the WE signals will be replicated into phase-specific **dfi_we_n_pN** signals that define the write enable for each phase N of the PHY clock. These signals are only applicable for DDR1, LPDDR1, DDR2 and DDR3 memory systems. For LPDDR2 memory systems, these signals must be held in the idle state.

a. This signal is not meaningful during initialization. Therefore, no default value is required to be maintained for this signal. For LPDDR2 memory systems, the **dfi_address** signal must be driven with a NOP until **dfi_init_complete** is asserted.

b. This signal is not meaningful during initialization. Therefore, no default value is required to be maintained for this signal.

c. Most memory devices define CKE as low at reset. However, some devices, such as Mobile DDR, define CKE as high at reset. The default value should reflect the memory definition.

TABLE 3. *Control Timing Parameter*

Parameter	Defined By	Min	Max	Unit	Description
$\mathbf{t_{ctrl\_delay}}$	PHY	0	- [a]	DFI clock cycles	Specifies the number of DFI clock cycles after an assertion or de-assertion of the DFI control signals that the control signals at the PHY-DRAM interface reflect the assertion or de-assertion. If the DFI clock and the memory clock are not phase-aligned, this timing parameter should be rounded up to the next integer value.

a. The DFI does not specify a maximum value. The range of values supported is implementation-specific.

3.2 Write Data Interface

The write data interface handles the transmission of write data across the DFI. The write mechanism defined by the DFI includes signal definitions along with timing relationships defined by DFI timing parameters. The signals **dfi_wrdata**, **dfi_wrdata_en**, **dfi_wrdata_mask** along with the related timing parameters $\mathbf{t_{phy\_wrlat}}$ and $\mathbf{t_{phy\_wrdata}}$ and $\mathbf{t_{phy\_wrdelay}}$ are described in Table 4 and Table 5.

The **dfi_wrdata_en** signal is asserted $\mathbf{t_{phy\_wrlat}}$ cycles after a write command is asserted on the DFI control interface and must remain asserted for the number of contiguous cycles that write data will be sent. The **dfi_wrdata** stream will begin $\mathbf{t_{phy\_wrdata}}$ cycles after the **dfi_wrdata_en** signal is asserted. The **dfi_wrdata_mask** signal follows the same timing as the **dfi_wrdata** signal, $\mathbf{t_{phy\_wrdata}}$ cycles after the **dfi_wrdata_en** signal is asserted.

The $\mathbf{t_{phy\_wrlat}}$ parameter defines the number of cycles between when the write command is sent on the DFI to assertion of the **dfi_wrdata_en** signal. This is a PHY-defined parameter, but may be specified in terms of other fixed system values. The **dfi_wrdata_en** signal must be asserted based on this timing parameter.

The $\mathbf{t_{phy\_wrdata}}$ parameter defines the timing requirements between the assertion of the **dfi_wrdata_en** signal at the DFI boundary and when the write data is sent on the **dfi_wrdata** signal. The exact value of this parameter for a particular application is determined by how many cycles prior to write data that the PHY must receive the **dfi_wrdata_en** signal. In previous versions of the DFI specification, this value was fixed at 1 cycle. If the PHY requires notification of pending write data sooner, the $\mathbf{t_{phy\_wrdata}}$ parameter may be increased. However, setting $\mathbf{t_{phy\_wrdata}}$ to a value greater than 1 may restrict the minimum write latency supported by the interface. The DFI specification does not dictate a value but does require that once this value has been determined, the **dfi_wrdata** signal must be asserted based on this timing parameter.

These timing parameters ($\mathbf{t_{phy\_wrlat}}$ and $\mathbf{t_{phy\_wrdata}}$) must be held constant while commands are being executed on the DFI bus; however, if necessary, these values may be changed when the bus is idle. The two timing parameters $\mathbf{t_{phy\_wrdata}}$ and $\mathbf{t_{phy\_wrlat}}$ work together to define the number of cycles from the assertion of a write command on the DFI control interface to when write data is driven on the DFI bus.

For frequency ratio systems, the signals will be replicated into phase-specific signals with a suffix of "_pN" that defines the signal value for each phase N of the PHY clock. Phase 0 may exclude the suffix if desired. An additional timing parameter $\mathbf{t_{phy\_wrdelay}}$ also exists for frequency ratio systems. This parameter is used to delay the write data in the PHY relative to the enable signal to define an accurate enable/data timing relationship. This timing parameter is included so that the MC may always maintain an alignment of the write data stream within a DFI data word beginning with phase 0. The MC may optionally send the data unaligned and not utilize this timing parameter.

More information on the write data interface is provided in Section 4.3, “Write Transactions”. The signals and parameter in the write data interface are listed in Table 4 and Table 5.

TABLE 4. *Write Data Signals*

Signal	From	Width	Default	Description
dfi_wrdata or **dfi_wrdata_pN**	MC	DFI Data Width	- [a]	Write data bus. The write data stream must begin $\mathbf{t_{phy\_wrdata}}$ cycles after the **dfi_wrdata_en** signal is asserted for the number of cycles that the **dfi_wrdata_en** signal is asserted. For frequency ratio systems, the write data bus will be replicated into phase-specific **dfi_wrdata_pN** buses that define the write data for each phase N of the PHY clock.
dfi_wrdata_en or **dfi_wrdata_en_pN**	MC	DFI Data Enable Width [b]	0x0	Write data and data mask enable. These signals must be asserted $\mathbf{t_{phy\_wrdata}}$ cycles before the data and data mask are sent on the DFI interface. The **dfi_wrdata_en** signal must be sent $\mathbf{t_{phy\_wrlat}}$ cycles after the write command. Once the **dfi_wrdata_en** signal is asserted, it must remain asserted for the number of contiguous cycles of write data passed through the DFI write data interface. The width of the **dfi_wrdata_en** signal is defined as a DFI term. Ideally, there will be a single **dfi_wrdata_en** bit for each slice of memory data. The **dfi_wrdata_en** [0] signal corresponds with the lowest segment of **dfi_wrdata** signals. For frequency ratio systems, the write data enable signals will be replicated into phase-specific **dfi_wrdata_en_pN** signals that define the validity of write data for each phase N of the PHY clock.
dfi_wrdata_mask or **dfi_wrdata_mask_pN**	MC	DFI Data Width / 8	- [a]	Write data byte mask. The timing is the same as for the **dfi_wrdata** bus. The **dfi_wrdata_mask** [0] signal defines masking for the **dfi_wrdata** [7:0] signals, the **dfi_wrdata_mask** [1] signal defines masking for the **dfi_wrdata** [15:8] signals, etc. If the **dfi_wrdata** bus is not a multiple of 8, then the uppermost bit of the **dfi_wrdata_mask** signal corresponds to the most significant partial byte of data. For frequency ratio systems, the write data mask bus will be replicated into phase-specific **dfi_wrdata_mask_pN** buses that define the write data masking information for each phase N of the PHY clock.

a. This signal is not meaningful during initialization. Therefore, no default value is required to be maintained for this signal.

b. Since all bits of the **dfi_wrdata_en** signal are identical, the width of the signal on the MC side and the PHY side may be different; the PHY is not required to use all of the bits.

TABLE 5. *Write Data Timing Parameters*

Parameter	Defined By	Min	Max	Unit	Description
t_{phy_wrdata}	PHY	0	- [a]	DFI PHY clock cycles [b]	Specifies the number of DFI PHY clock cycles between when the **dfi_wrdata_en** signal is asserted to when the associated write data is driven on the **dfi_wrdata** signal. This has no impact on performance, only adjusts the relative time between enable and data transfer. DFI 1.0 and DFI 2.0 memory controllers will only support a t_{phy_wrdata} value of 1. A PHY will generally be designed to operate at a single t_{phy_wrdata} value; however, the MC should support a range of values. NOTE: This parameter may be specified as a fixed value, or as a constant based on other fixed values in the system.
$t_{phy\_wrdelay}$	PHY	0	1 or 3 [c]	DFI PHY clock cycles [b]	Specifies the number of DFI PHY clock cycles of additional delay that the PHY must insert between the write data enable and write data once data has been captured from the DFI bus. This parameter is only used in frequency ratio systems.
t_{phy_wrlat}	PHY	0	- [a]	DFI PHY clock cycles [b]	Specifies the number of DFI PHY clock cycles between when a write command is sent on the DFI control interface and when the **dfi_wrdata_en** signal is asserted. NOTE: This parameter may be specified as a fixed value, or as a constant based on other fixed values in the system.

a. The DFI does not specify a maximum value. The range of values supported is implementation-specific.

b. This timing parameter is defined in terms of DFI PHY clock cycles for frequency ratio systems. For matched frequency systems, a DFI PHY clock is identical to the DFI clock.

c. The PHY must support a maximum write data delay of one less than the clock ratio. A 1:2 frequency ratio PHY must support values of 0 and 1. A 1:4 frequency ratio PHY must support a range of 0-3.

3.3 Read Data Interface

The read data interface handles the return of read data across the DFI. The read mechanism defined by the DFI includes signal definitions along with timing relationships defined by DFI timing parameters. The signals **dfi_rddata**, **dfi_rddata_en**, **dfi_rddata_valid**, the LPDDR2 signal **dfi_rddata_dnv**, along with the related timing parameters t_{rddata_en} and t_{phy_rdlat} are described in Table 6 and Table 7.

The **dfi_rddata_en** signal is asserted t_{rddata_en} cycles after a read command is asserted on the DFI control interface and must remain asserted for the number of contiguous cycles that read data is expected. One continuous assertion of the **dfi_rddata_en** signal may encompass data for multiple read commands. The **dfi_rddata_en** signal de-asserts to signify there is no more contiguous data expected from the DFI read command(s). Note that the **dfi_rddata_en** signal is not required to be asserted for any fixed number of cycles.

The $\mathbf{t_{rddata\_en}}$ parameter defines the timing requirements between the read command on the DFI interface and the assertion of the **dfi_rddata_en** signal at the DFI boundary for the start of contiguous read data expected on the DFI interface. The exact value of this parameter for a particular application is determined by the components in the entire DRAM system. The DFI specification does not dictate a value but does require that once this value has been determined, the **dfi_rddata_en** signal must be asserted based on this timing parameter.

The $\mathbf{t_{phy\_rdlat}}$ parameter defines the maximum number of cycles allowed from the assertion of the **dfi_rddata_en** signal to the assertion of the **dfi_rddata_valid** signal. This parameter is specified by the system, but the exact value of this parameter is not determined by the DFI specification.

These timing parameters ($\mathbf{t_{rddata\_en}}$ and $\mathbf{t_{phy\_rdlat}}$) must be held constant while commands are being executed on the DFI bus; however, if necessary, these values may be changed when the bus is idle. These parameters work together to define a maximum number of cycles from the assertion of a read command on the DFI control interface to the assertion of the **dfi_rddata_valid** signal, indicating the first valid data of the contiguous read data. Read data may be returned earlier by asserting the **dfi_rddata_valid** signal before $\mathbf{t_{phy\_rdlat}}$ cycles have expired. When the signal **dfi_rddata_valid** is asserted, the entire DFI read data word must be valid. For the LPDDR2 DFI, the signal **dfi_rddata_dnv** must also be sent with the read data signal **dfi_rddata** when the **dfi_rddata_valid** signal is asserted.

For frequency ratio systems, the read data enable signal will be replicated into phase-specific signals with a suffix of "_pN" that defines the signal value for each phase N of the DFI PHY clock relative to the DFI clock. The read data, read data valid and read data not valid signals will be replaced with DFI data word-specific signals with a suffix of "_wN" with the DFI data word N. For all signal types, the suffix for phase 0/word 0 is optional.

More information on the read data interface is provided in Section 4.4, "Read Transactions". The signals and parameters in the read data interface are listed in Table 6 and Table 7.

TABLE 6. *Read Data Signals*

Signal	From	Width	Default	Description
dfi_rddata or **dfi_rddata_wN**	PHY	DFI Data Width	- [a]	Read data bus. Read data is expected to be received at the MC within $\mathbf{t_{phy\_rdlat}}$ cycles after the **dfi_rddata_en** signal is asserted. For frequency ratio systems, the read data bus will be replicated into DFI data word-specific **dfi_rddata_wN** buses that define the read data for each DFI data word. For a 1:2 frequency ratio system, there are 2 **dfi_rddata_wN** signals. For a 1:4 frequency ratio system, there are 4 **dfi_rddata_wN** signals.

TABLE 6. *Read Data Signals*

Signal	From	Width	Default	Description
dfi_rddata_en or **dfi_rddata_en_pN**	MC	DFI Data Enable Width [b]	0x0	Read data enable. The **dfi_rddata_en** signal must be asserted t_{rddata_en} cycles after the assertion of a read command on the DFI control interface and remains valid for the duration of contiguous read data expected on the **dfi_rddata** bus. The width of the **dfi_rddata_en** signal is defined as a DFI term. Ideally, there will be a single **dfi_rddata_en** bit for each slice of memory data. The **dfi_rddata_en** [0] signal corresponds with the lowest segment of **dfi_rddata** signals. For frequency ratio systems, the read data enable signals will be replicated into phase-specific **dfi_rddata_en_pN** signals that define the read data enable for each phase N of the PHY clock.
dfi_rddata_valid or **dfi_rddata_valid_wN**	PHY	DFI Read Data Valid Width [c]	0x0	Read data valid indicator. The **dfi_rddata_valid** signal will be asserted with the read data for the number of cycles that data is being sent. The timing is the same as for the **dfi_rddata** bus. For frequency ratio systems, the read data valid signals will be replicated into DFI data word-specific **dfi_rddata_valid_wN** signals that define the validity of the read data. For a 1:2 frequency ratio system, there are 2 **dfi_rddata_valid_wN** signals. For a 1:4 frequency ratio system, there are 4 **dfi_rddata_valid_wN** signals.
dfi_rddata_dnv or **dfi_rddata_dnv_wN**	PHY	DFI Data Width / 8	0x0	DFI data not valid. The timing is the same as for the **dfi_rddata_valid** signal. The **dfi_rddata_dnv** [0] signal correlates to the **dfi_rddata** [7:0] signals, the **dfi_rddata_dnv** [1] signal correlates to the **dfi_rddata** [15:8] signals, etc. If the **dfi_rddata** bus is not a multiple of 8, then the uppermost bit of the **dfi_rddata_dnv** signal corresponds to the most significant partial byte of data. For frequency ratio systems, the read data not valid signals will be replicated into DFI data word-specific **dfi_rddata_dnv_wN** signals that define the validity of read data. For a 1:2 frequency ratio system, there are 2 **dfi_rddata_dnv_wN** signals. For a 1:4 frequency ratio system, there are 4 **dfi_rddata_dnv_wN** signals. This signal is only required for DFI LPDDR2 support.

a. This signal is not meaningful during initialization. Therefore, no default value is required to be maintained for this signal.

b. Since all bits of the **dfi_rddata_en** signal are identical, the width of the signal on the MC side and the PHY side may be different; the PHY is not required to use all of the bits.

c. Since all bits of the **dfi_rddata_valid** signal are identical, the width of the signal on the MC side and the PHY side may be different; the MC is not required to use all of the bits.

TABLE 7. *Read Data Timing Parameters*

Parameter	Defined By	Min	Max	Unit	Description
t_{phy_rdlat}	PHY	0	- [a]	DFI PHY clock cycles [b]	Specifies the maximum number of DFI PHY clock cycles allowed from the assertion of the **dfi_rddata_en** signal to the assertion of the **dfi_rddata_valid** signal. NOTE: This parameter may be specified as a fixed value, or as a constant based on other fixed values in the system.
t_{rddata_en}	System	0	- [a]	DFI PHY clock cycles [b]	Specifies the number of DFI PHY clock cycles from the assertion of a read command on the DFI to the assertion of the **dfi_rddata_en** signal. NOTE: This parameter may be specified as a fixed value, or as a constant based on other fixed values in the system.

a. The DFI does not specify a maximum value. The range of values supported is implementation-specific.

b. This timing parameter is defined in terms of DFI PHY clock cycles for frequency ratio systems. For matched frequency systems, a DFI PHY clock is identical to the DFI clock.

3.4 Update Interface

During system operation, the system may require updates to internal settings to compensate for environmental conditions. To ensure that updates do not interfere with signals on the DRAM interface, the DFI supports update modes where the DFI read, write, and control interface are suspended from normal activity. The DFI specification supports both MC-initiated and PHY-initiated updates. More information on the update interface is provided in Section 4.5, “PHY Update”.

If a MC initiates an update request by asserting the **dfi_ctrlupd_req** signal, the request can be acknowledged or ignored by the PHY. If the request is acknowledged by asserting the **dfi_ctrlupd_ack** signal, the protocol described in Section 4.5.1, “MC-Initiated Update” must be followed. The DFI specification requires the MC to issue update requests and it specifies a maximum interval ($t_{ctrlupd\_interval}$) at which requests must be offered. The MC should assert the **dfi_ctrlupd_req** signal at the end of memory initialization to signify that the initialization is complete.

If a PHY initiates an update request by asserting the **dfi_phyupd_req** signal, the request must be acknowledged by the MC through a **dfi_phyupd_ack** signal assertion. The DFI specifies up to 4 different update PHY-initiated request modes. Each mode differs only in the number of cycles that the DFI interface must be suspended while the update occurs. During this time, the MC is responsible for placing the system in a state where the DFI bus is suspended from all activity other than activity specifically related to the update process being executed. Refer to Section 4.5.2, “PHY-Initiated Update” for more details on this protocol. The DFI specification does not require the PHY to issue update requests nor does it specify an interval in which requests must be offered. If the PHY chooses to offer update requests, it must follow the specified protocol.

It is possible that both update request signals (**dfi_phyupd_req** and **dfi_ctrlupd_req**) could be asserted at the same time. When both request signals are driven, either the PHY or the MC could acknowledge the other's request. In this case, the request that was not acknowledged could be de-asserted. This is the only situation in which the PHY is permitted to de-assert the **dfi_phyupd_req** signal without an acknowledge from the MC. The acknowledged request must follow the appropriate protocol.

The signals and timing parameters in the update interface are listed in Table 8 and Table 9.

TABLE 8. *Update Interface Signals*

Signal	From	Width	Default	Description
dfi_ctrlupd_ack	PHY	1 bit	0x0	MC-initiated update acknowledge. The **dfi_ctrlupd_ack** signal is asserted to acknowledge a MC-initiated update request. The PHY is not required to acknowledge this request. While this signal is asserted, the DFI bus must remain idle other than any transactions specifically associated with the update process. If the PHY chooses to acknowledge the request, the **dfi_ctrlupd_ack** signal must be asserted before the **dfi_ctrlupd_req** signal de-asserts. If the PHY chooses to ignore the request, the **dfi_ctrlupd_ack** signal must remain de-asserted until the **dfi_ctrlupd_req** signal is de-asserted. The **dfi_ctrlupd_req** signal is guaranteed to be asserted for at least $\mathbf{t_{ctrlupd\_min}}$ cycles.
dfi_ctrlupd_req	MC	1 bit	0x0	MC-initiated update request. The **dfi_ctrlupd_req** signal is used with a MC-initiated update to indicate that the DFI will be idle for some time, in which the PHY may perform an update. The **dfi_ctrlupd_req** signal must be asserted for a minimum of $\mathbf{t_{ctrlupd\_min}}$ cycles and a maximum of $\mathbf{t_{ctrlupd\_max}}$ cycles. A **dfi_ctrlupd_req** signal assertion is an invitation for the PHY to update and does not require a response. The behavior of the **dfi_ctrlupd_req** signal is dependent on the **dfi_ctrlupd_ack** signal: • If the update is acknowledged by the PHY, then the **dfi_ctrlupd_req** signal will remain asserted as long as the **dfi_ctrlupd_ack** signal asserted, but will de-assert before $\mathbf{t_{ctrlupd\_max}}$ expires. While this signal is asserted, the DFI bus will remain idle other than any transactions specifically associated with the update process. • If the update is not acknowledged, the **dfi_ctrlupd_req** signal may de-assert at any time after $\mathbf{t_{ctrlupd\_min}}$, and before $\mathbf{t_{ctrlupd\_max}}$.

TABLE 8. *Update Interface Signals*

Signal	From	Width	Default	Description
dfi_phyupd_ack	MC	1 bit	0x0	PHY-initiated update acknowledge. The **dfi_phyupd_ack** signal is used for a PHY-initiated update to indicate that the DFI is idle and will remain so until the **dfi_phyupd_req** signal de-asserts. The MC must assert the **dfi_phyupd_ack** signal within $\mathbf{t_{phyupd\_resp}}$ cycles of the **dfi_phyupd_req** signal, and must remain asserted as long as the **dfi_phyupd_req** signal remains asserted. The **dfi_phyupd_ack** signal must de-assert on the cycle following the **dfi_phyupd_req** signal de-assertion. While this signal is asserted, the DFI bus must remain idle other than any transactions specifically associated with the update process. The entire time period from when the **dfi_phyupd_ack** signal is asserted to when the **dfi_phyupd_req** signal is de-asserted will be a maximum of $\mathbf{t_{phyupd\_typeX}}$ cycles, based on the **dfi_phyupd_type** signal.
dfi_phyupd_req	PHY	1 bit	0x0	PHY-initiated update request. The **dfi_phyupd_req** signal is used for a PHY-initiated update to indicate that the PHY requires the DFI to not send control, read or write commands or data for a specified period of time. The maximum time required is specified by the $\mathbf{t_{phyupd\_typeX}}$ parameter associated with the **dfi_phyupd_type** signal. Once asserted, the **dfi_phyupd_req** signal must remain asserted until the request is acknowledged by the assertion of the **dfi_phyupd_ack** signal and the update has been completed. The MC must acknowledge this request. While this signal is asserted, the DFI bus must remain idle other than any transactions specifically associated with the update process. The de-assertion of the **dfi_phyupd_req** signal triggers the de-assertion of the **dfi_phyupd_ack** signal.
dfi_phyupd_type	PHY	2 bits	- [a]	PHY-initiated update select. The **dfi_phyupd_type** signal indicates which one of the 4 types of PHY update times is being requested by the **dfi_phyupd_req** signal. The value of the **dfi_phyupd_type** signal will determine which of the timing parameters ($\mathbf{t_{phyupd\_type0}}$, $\mathbf{t_{phyupd\_type1}}$, $\mathbf{t_{phyupd\_type2}}$, $\mathbf{t_{phyupd\_type3}}$) is relevant. The **dfi_phyupd_type** signal must remain constant during the entire time the **dfi_phyupd_req** signal is asserted.

a. This signal is not meaningful during initialization. Therefore, no default value is required to be maintained for this signal.

TABLE 9. *Update Timing Parameters*

Parameter	Defined By	Min	Max	Unit	Description
$t_{ctrlupd\_interval}$	MC	- [a]	- [b]	DFI clock cycles	Specifies the maximum number of DFI clock cycles that the MC may wait between assertions of the **dfi_ctrlupd_req** signal.
$t_{ctrlupd\_min}$	MC	1	- [b]	DFI clock cycles	Specifies the minimum number of DFI clock cycles that the **dfi_ctrlupd_req** signal must be asserted.
$t_{ctrlupd\_max}$	MC	- [a]	- [b]	DFI clock cycles	Specifies the maximum number of DFI clock cycles that the **dfi_ctrlupd_req** signal can assert.
t_{phyupd_type0}	PHY	1	- [b]	DFI clock cycles	Specifies the maximum number of DFI clock cycles that the **dfi_phyupd_req** signal may remain asserted after the assertion of the **dfi_phyupd_ack** signal for **dfi_phyupd_type** = 0x0. The **dfi_phyupd_req** signal may de-assert at any cycle after the assertion of the **dfi_phyupd_ack** signal.
t_{phyupd_type1}	PHY	1	- [b]	DFI clock cycles	Specifies the maximum number of DFI clock cycles that the **dfi_phyupd_req** signal may remain asserted after the assertion of the **dfi_phyupd_ack** signal for **dfi_phyupd_type** = 0x1. The **dfi_phyupd_req** signal may de-assert at any cycle after the assertion of the **dfi_phyupd_ack** signal.
t_{phyupd_type2}	PHY	1	- [b]	DFI clock cycles	Specifies the maximum number of DFI clock cycles that the **dfi_phyupd_req** signal may remain asserted after the assertion of the **dfi_phyupd_ack** signal for **dfi_phyupd_type** = 0x2. The **dfi_phyupd_req** signal may de-assert at any cycle after the assertion of the **dfi_phyupd_ack** signal.
t_{phyupd_type3}	PHY	1	- [b]	DFI clock cycles	Specifies the maximum number of DFI clock cycles that the **dfi_phyupd_req** signal may remain asserted after the assertion of the **dfi_phyupd_ack** signal for **dfi_phyupd_type** = 0x3. The **dfi_phyupd_req** signal may de-assert at any cycle after the assertion of the **dfi_phyupd_ack** signal.
t_{phyupd_resp}	PHY	1	- [b]	DFI clock cycles	Specifies the maximum number of DFI clock cycles after the assertion of the **dfi_phyupd_req** signal to the assertion of the **dfi_phyupd_ack** signal.

a. The DFI does not specify a minimum value. The range of values supported is an implementation-specific design parameter.

b. The DFI does not specify a maximum value. The range of values supported is an implementation-specific design parameter.

3.5 Status Interface

The DFI requires status information for initialization and clock control to the DRAM devices. These signals are used to convey information between the MC and the PHY. An optional frequency change handshaking protocol is included in this interface.

The **dfi_init_start** signal and the **dfi_init_complete** signal are used at initialization and as part of the frequency change protocol. At initialization, the **dfi_init_start** signal indicates the validity of the **dfi_freq_ratio** and/or **dfi_data_byte_disable** signals. During normal operation, once both the **dfi_init_complete** and **dfi_init_start** signals have been asserted, the **dfi_init_start** signal is used to trigger frequency change. At initialization, the **dfi_init_complete** signal indicates that the PHY is ready to accept DFI transactions. During frequency change, this signal indicates an acceptance of the frequency change request.

The optional **dfi_data_byte_disable** signal is used to inform the PHY which data slices will be unused. The value of the **dfi_data_byte_disable** signal must be defined at initialization and is not expected to change.

The byte lanes correspond to the byte order defined for the **dfi_wrdata** and **dfi_rddata** signals. The MC and the PHY must both support the same partial use of data signals in order to use byte disabling. For example, if the MC disables the upper bits (most significant bits) of the data bus, the PHY must be able to operate accurately with the remaining byte lanes. The DFI specification does not define supported active/inactive patterns, and therefore care must be taken to insure the interoperability of the MC and the PHY.

The **dfi_dram_clk_disable** signal is used by the MC to inform the PHY when to enable/disable the clock to the memory devices. The timing parameters $\mathbf{t_{dram\_clk\_disable}}$ and $\mathbf{t_{dram\_clk\_enable}}$ define the timing of the DRAM clock enable/disable relative to this signal.

The optional **dfi_freq_ratio** signal is used to convey frequency ratio information to the PHY. This static signal indicates the ratio expected by the MC, and will dictate how control, read and write information is passed across the DFI.

The optional DIMM support signals are provided specifically for DDR3 memory systems. These DIMMs include additional I/O pins with information that should be communicated between the memory devices and the MC. The MC communicates its command parity setting on the **dfi_parity_in** signal and the PHY reports command parity errors on the **dfi_parity_error** signal.

More information on initialization is provided in Section 4.1, “Initialization”, more information on the clock disable interface is provided in Section 4.6, “DFI Clock Disabling”, more information on the frequency ratio protocol is provided in Section 4.7, “Frequency Ratios Across the DFI” and more information on the frequency change protocol is provided in Section 4.8, “Frequency Changing”.

The signals and timing parameters for the status interface are listed in Table 10 and Table 11.

TABLE 10. *Status Interface Signals*

Signal	From	Width	Default	Description
dfi_data_byte_disable	MC	DFI Data Width / 8	- [a]	Data byte disable. When set, this signal indicates that the associated data byte is not being used. In this state, the PHY is permitted to place the associated bytes in a low power state. This signal is optional. When the bit is clear, the byte will operate normally. The byte lanes correspond to the byte order defined for the **dfi_wrdata** and **dfi_rddata** signals. This is a static signal and may only be defined at initialization. Once defined, the MC will drive the **dfi_init_start** signal to the PHY.
dfi_dram_clk_disable	MC	DFI Chip Select Width	0x0 [b]	DRAM clock disable. When active, this indicates to the PHY that the clocks to the DRAM devices must be disabled such that the clock signals hold a constant value. When the **dfi_dram_clk_disable** signal is inactive, the DRAMs should be clocked normally.
dfi_freq_ratio	MC	2 bits	- [a]	DFI frequency ratio indicator. This signal defines the frequency ratio for this system. This signal is only required when using the frequency ratio protocol. The **dfi_freq_ratio** signal must be defined at initialization and must be held constant during operation. Once defined, the MC will drive the **dfi_init_start** signal to the PHY. • ‘b00 = 1:1 MC:PHY frequency ratio (matched frequency) • ‘b01 = 1:2 MC:PHY frequency ratio • ‘b10 = 1:4 MC:PHY frequency ratio • ‘b11 = Reserved
dfi_init_complete	PHY	1 bit	0x0	PHY initialization complete. The **dfi_init_complete** signal indicates that the PHY is able to respond to any proper stimulus on the DFI. All DFI signals that communicate commands or status must be held at their default values until the **dfi_init_complete** signal asserts. During a PHY re-initialization request (such as a frequency change), this signal will be de-asserted. For a frequency change request, the de-assertion of the **dfi_init_complete** signal acknowledges the frequency change protocol. Once de-asserted, the signal should only be re-asserted within $\mathbf{t_{init\_complete}}$ cycles after the **dfi_init_start** signal has de-asserted, and once the PHY has completed re-initialization.

TABLE 10. *Status Interface Signals*

Signal	From	Width	Default	Description
dfi_init_start	MC	1 bit	0x0 [c] 0x1	DFI initialization start. When this signal is asserted during initialization, the MC is indicating that the **dfi_data_byte_disable** and/or **dfi_freq_ratio** signals have been defined. When this signal is asserted during normal operation, the MC is requesting a frequency change. This signal is optional and will only be relevant for MCs and PHYs that support data byte disabling, frequency ratio or the frequency change protocol. During initialization, a **dfi_init_start** signal assertion is an indication to the PHY that the frequency ratio and data byte lane use has been defined. The PHY may use this signal to know when these settings are valid from the MC. The MC will hold this signal until the **dfi_init_complete** signal has been asserted by the PHY, after which it may release the **dfi_init_start** signal. During normal operation, once both the **dfi_init_start** and **dfi_init_complete** signals have been asserted, a **dfi_init_start** signal assertion is an invitation for the PHY to accept frequency changing and does not require a response. However, if desired, the PHY must respond within $\mathbf{t_{init\_start}}$ cycles, or the opportunity for frequency change will be withdrawn until the MC re-asserts this signal. If the **dfi_init_start** signal is asserted, the MC and the PHY must reset their DFI read data word pointers to 0. Refer to Section 4.7.2.2, “Read Data Interface” for more information. The behavior of the **dfi_init_start** signal is dependent on the **dfi_init_complete** signal: • If the PHY wishes to accept the frequency change request, it must de-assert the **dfi_init_complete** signal within $\mathbf{t_{init\_start}}$ cycles of the **dfi_init_start** assertion. The MC will continue to hold the **dfi_init_start** signal asserted until the clock frequency change has been completed. The de-assertion should be used by the PHY to re-initialize on the new clock frequency. • If the frequency change is not acknowledged (the **dfi_init_complete** signal remains asserted), the **dfi_init_start** signal must de-assert after $\mathbf{t_{init\_start}}$ cycles.
dfi_parity_error	PHY	1 bit	0x0	Parity error indicator. This signal will be driven when a parity error is detected in any of the DIMMs for the command. This signal correlates to the DIMM output signal ERR_OUT#. This signal is optional and will only be relevant for MCs and PHYs that support command parity and DIMMs with this output. The **dfi_parity_error** signal will be received at the MC within $\mathbf{t_{phy\_paritylat}}$ cycles after the command associated with a **dfi_parity_in** signal is driven.

TABLE 10. *Status Interface Signals*

Signal	From	Width	Default	Description
dfi_parity_in	MC	1 bit	0x0 [d] 0x1 [b]	Parity setting. This signal will be sent with each DFI command and will be valid for 1 cycle. This setting applies to the **dfi_address**, **dfi_bank**, **dfi_cas_n**, **dfi_ras_n** and **dfi_we_n** signals and correlates to the DIMM input signal PAR_IN. This signal is optional and will only be relevant for PHYs that support command parity and require the MC to generate the parity information. • 'b0 = An even number of these control interface signals are electrically high. • 'b1 = An odd number of these control interface signals are electrically high.

a. At initialization, this signal must be driven with the valid settings for the system to convey information to the PHY. Refer to Section 4.1, "Initialization".

b. This signal may be programmed while **dfi_init_complete** is not asserted. It should be programmed according to which clocks are being used.

c. The PHY may optionally wait for the **dfi_init_start** signal assertion before asserting the **dfi_init_complete** signal. If

d. The value of this signal must reflect the correct parity for the selected control interface signals.

TABLE 11. *Status Timing Parameters*

Parameter	Defined By	Min	Max	Unit	Description
$t_{dram\_clk\_disable}$	PHY	0	- [a]	DFI clock cycles	Specifies the number of DFI clock cycles from the assertion of the **dfi_dram_clk_disable** signal on the DFI until the clock to the DRAM memory devices, at the PHY-DRAM boundary, maintains a low value. If the DFI clock and the memory clock are not phase-aligned, this timing parameter should be rounded up to the next integer value. NOTE: This parameter may be specified as a fixed value, or as a constant based on other fixed values in the system.
$t_{dram\_clk\_enable}$	PHY	0	- [a]	DFI clock cycles	Specifies the number of DFI clock cycles from the de-assertion of the **dfi_dram_clk_disable** signal on the DFI until the first valid rising edge of the clock to the DRAM memory devices, at the PHY-DRAM boundary. If the DFI clock and the memory clock are not phase-aligned, this timing parameter should be rounded up to the next integer value. NOTE: This parameter may be specified as a fixed value, or as a constant based on other fixed values in the system.
$t_{init\_complete}$	PHY	0	- [a]	DFI clock cycles	Specifies the maximum number of DFI clock cycles after the de-assertion of the **dfi_init_start** signal to the re-assertion of the **dfi_init_complete** signal.

TABLE 11. *Status Timing Parameters*

Parameter	Defined By	Min	Max	Unit	Description
t_{init_start}	MC	0	- [a]	DFI clock cycles	Specifies the number of DFI clock cycles from the assertion of the **dfi_init_start** signal on the DFI until the PHY must respond by de-asserting the **dfi_init_complete** signal. If the **dfi_init_complete** signal is not de-asserted within this time period, the PHY is indicating that it can not, or does not wish to, support the frequency change. At this point, the MC must abort the request and release the **dfi_init_start** signal. Once t_{init_start} expires, the PHY must not de-assert the **dfi_init_complete** signal. The MC may re-assert **dfi_init_start** at a later point.
$t_{phy\_paritylat}$	PHY	4	- [a]	DFI clock cycles	Specifies the maximum number of DFI clock cycles between when the **dfi_parity_in** signal is driven and when the associated **dfi_parity_error** is returned.

a. The DFI does not specify a maximum value. The range of values supported is an implementation-specific design parameter.

3.6 Training Interface

DDR3 and LPDDR2 memories feature additional functions which allow for more accurate alignment of critical timing signals. The DFI specification accounts for these functions by providing a training interface.

The DFI specification supports read leveling and write leveling. Read leveling is used by both DDR3 and LPDDR2 memory systems while write leveling is a specific function of DDR3 memory systems. Read leveling encompasses both data eye training and gate training. More information on the training interface is provided in Section 4.10, “Training Operations - Read and Write Leveling”. The signals and timing parameters for the training interface are listed in Table 12 and Table 13.

Not all DFI training signals are used in all systems. The **dfi_rdlvl_mode** and **dfi_rdlvl_gate_mode** signals are required for all LPDDR2 memory systems. The **dfi_rdlvl_mode**, **dfi_wrlvl_mode** and **dfi_rdlvl_gate_mode** signals are required for all DDR3 memory systems. These signals are used to indicate the type of data eye training, gate training and write leveling supported by the PHY: “No Support”, “MC Evaluation”, “PHY Evaluation” or “PHY Independent”. The MC must support all of the leveling modes to be fully DFI-compliant; however, the PHY is expected to support only a single mode per training operation. The signals required for data eye training, gate training and write leveling must be limited to the signals defined in this specification. The signal set for the training interface is mode-dependent and the relevance of each signal is indicated in the descriptions.

For PHY Evaluation mode, it is possible to perform both data eye training and gate training using just the read leveling signals since these operations result in identical sequences for the MC. However, a separate set of signals is provided for gate training

for MC Evaluation mode and may be used for gate training by a PHY operating in PHY Evaluation mode if desired.

The read and write leveling signals that communicate from the MC to the PHY are internally fanned out inside the MC to allow a direct connection from the MC to each PHY memory data slice. Other than the delay signals (**dfi_rdlvl_delay_X**, **dfi_rdlvl_gate_delay_X** and **dfi_wrlvl_delay_X**), all of these fanout signals originating from the MC to the PHY must be driven with the same value. The read and write leveling signals that communicate from the PHY to the MC may be individually driven by each memory data slice or collectively driven as a single signal.

TABLE 12. *Training Interface Signals*

Signal	From	Width	Default	Description
dfi_rdlvl_req	PHY	DFI Read Leveling PHY IF Width	0x0	PHY-initiated data eye training request. This is an optional signal for the PHY; other sources may be used to initiate data eye training or the MC may initiate data eye training independently. The PHY may drive independent data eye training requests from each data slice; however the MC must read level all data slices based on a single assertion of the **dfi_rdlvl_req** signal. If the PHY asserts the **dfi_rdlvl_req** signal, the MC must acknowledge the request by asserting the **dfi_rdlvl_en** signal within $\mathbf{t_{rdlvl\_resp}}$ cycles, after which the PHY should de-assert the **dfi_rdlvl_req** signal. The PHY should not assert this signal during initialization or a frequency change operation. The MC is responsible for any data eye training required during these operations. This signal is only applicable for MCs connecting to PHYs operating in MC Evaluation or PHY Evaluation modes for data eye training.
dfi_rdlvl_gate_req	PHY	DFI Read Leveling PHY IF Width	0x0	PHY-initiated gate training request. This is an optional signal for the PHY; other sources may be used to initiate gate training or the MC may initiate gate training independently. The PHY may drive independent gate training requests from each data slice; however the MC must gate train all data slices based on a single assertion of the **dfi_rdlvl_gate_req** signal. If the PHY asserts the **dfi_rdlvl_gate_req** signal, the MC must acknowledge the request by asserting the **dfi_rdlvl_gate_en** signal within $\mathbf{t_{rdlvl\_resp}}$ cycles, after which the PHY should de-assert the **dfi_rdlvl_gate_req** signal. The PHY should not assert this signal during initialization or a frequency change operation. The MC is responsible for any gate training required during these operations. This signal is only applicable for MCs connecting to PHYs operating in MC Evaluation or PHY Evaluation modes for gate training.

TABLE 12. *Training Interface Signals*

Signal	From	Width	Default	Description
dfi_rdlvl_mode	PHY	2 bits	- [a]	Data eye training support. Defines responsibility over the data eye training operation. The MC is required to support all of these modes. • 'b00 = Data eye training is not supported by the PHY. • 'b01 = MC Evaluation mode. The MC will enable and disable the data eye training logic in the PHY, analyze the results and adjust the delays. • 'b10 = PHY Evaluation mode. The MC enables and disables the data eye training logic in the PHY. The PHY contains logic to evaluate the results and set new delay values. • 'b11 = PHY Independent mode. The PHY performs all data eye training operations. This signal is required for all systems that support data eye training.
dfi_rdlvl_gate_mode	PHY	2 bits	- [a]	Gate training support. Defines responsibility over the gate training operation. The MC is required to support all of these modes. • 'b00 = Gate training is not supported by the PHY. • 'b01 = MC Evaluation mode. The MC will enable and disable the gate training logic in the PHY, analyze the results and adjust the delays. • 'b10 = PHY Evaluation mode. The MC enables and disables the gate training logic in the PHY. The PHY contains logic to evaluate the results and set new delay values. • 'b11 = PHY Independent mode. The PHY performs all gate training operations. This signal is required for all systems that support gate training.
dfi_rdlvl_en	MC	DFI Read Leveling MC IF Width	0x0	PHY data eye training logic enable. This signal is asserted during data eye training. If the PHY initiated the data eye training request (**dfi_rdlvl_req**), then this serves as an acknowledge of that request. • 'b0 = Normal operation • 'b1 = Data eye training logic enabled. The assertion of this signal immediately triggers data eye training. If the **dfi_rdlvl_en** signal is asserted, the MC and the PHY must reset their DFI read data word pointers to 0 on de-assertion of this signal. Refer to Section 4.7.2.2, "Read Data Interface" for more information. This signal is only applicable for MCs connecting to PHYs operating in MC Evaluation or PHY Evaluation modes for data eye training.

TABLE 12. *Training Interface Signals*

Signal	From	Width	Default	Description
dfi_rdlvl_gate_en	MC	DFI Read Leveling MC IF Width	0x0	PHY gate training logic enable. This signal is asserted during gate training. If the PHY initiated the gate training request (**dfi_rdlvl_gate_req**), then this serves as an acknowledge of that request. • 'b0 = Normal operation • 'b1 = Gate training logic enabled. The assertion of this signal immediately triggers gate training. If the **dfi_rdlvl_en** signal is asserted, the MC and the PHY must reset their DFI read data word pointers to 0 on de-assertion of this signal. Refer to Section 4.7.2.2, "Read Data Interface" for more information. This signal is only applicable for MCs connecting to PHYs operating in MC Evaluation or PHY Evaluation modes for gate training.
dfi_rdlvl_cs_n	MC	DFI Chip Select Width	_ b	Read leveling chip select. Indicates which chip select is currently active for either read leveling operation. This signal is only applicable for MCs connecting to PHYs operating in MC Evaluation or PHY Evaluation modes for read leveling.
dfi_rdlvl_edge	MC	DFI Read Leveling MC IF Width	_ b	Read leveling read DQS edge. Indicates which edge of the read DQS is currently being used for the read leveling sequence. This signal must remain constant throughout the sequence. It is not a requirement for the PHY to support read leveling of both the positive and negative edges of the read DQS. • 'b0 = Positive edge • 'b1 = Negative edge This signal is only applicable for MCs connecting to PHYs operating in MC Evaluation or PHY Evaluation modes for read leveling.
dfi_rdlvl_delay_X	MC	DFI Read Leveling Delay Width	_ b	Data eye training delay value. Indicates the programming of the PHY delay for read DQS sampling read data. The width of the **dfi_rdlvl_delay_X** signals is defined as a DFI term. In general, each memory data slice will be uniquely leveled and therefore a separate **dfi_rdlvl_delay_X** signal should be sent to each memory data slice X where **dfi_rdlvl_delay_0** corresponds to the first data slice. In some applications, the PHY may only use a subset of the delay signals provided by the MC. The width of each **dfi_rdlvl_delay_X** signal is defined by the programmability of the delay line. In some applications, the PHY may only use a subset of the delay signals provided by the MC. This signal is only applicable for MCs connecting to PHYs operating in MC Evaluation mode for data eye training.

TABLE 12. *Training Interface Signals*

Signal	From	Width	Default	Description
dfi_rdlvl_gate_delay_X	MC	DFI Read Leveling Gate Delay Width	- [b]	Gate training delay value. Indicates the programming of the PHY delay for the gate sampling read data. The width of the **dfi_rdlvl_gate_delay_X** signals is defined as a DFI term. In general, each memory data slice will be uniquely leveled and therefore a separate **dfi_rdlvl_gate_delay_X** signal should be sent to each memory data slice X where **dfi_rdlvl_gate_delay_0** corresponds to the first data slice. In some applications, the PHY may only use a subset of the delay signals provided by the MC. The width of each **dfi_rdlvl_gate_delay_X** signal is defined by the programmability of the delay line. In some applications, the PHY may only use a subset of the delay signals provided by the MC. This signal is only applicable for MCs connecting to PHYs operating in MC Evaluation mode for gate training.
dfi_rdlvl_load	MC	DFI Read Leveling MC IF Width	0x0	Read leveling load. The MC must send a one-cycle pulse on this signal when it has updated any of the delay times (**dfi_rdlvl_delay_X** or **dfi_rdlvl_gate_delay_X**) for the next read leveling command. This signal is only applicable for MCs connecting to PHYs operating in MC Evaluation mode for read leveling.
dfi_rdlvl_resp	PHY	DFI Read Leveling Response Width	- [b]	Read leveling response. Response definition depends on the mode of operation and the memory type for the system: • PHY Evaluation mode for DDR3 or LPDDR2 memory systems: The response indicates that the PHY has completed data eye training or gate training and centered the DQS relative to the data or placed the gate within the DQS preamble. • MC Evaluation mode for DDR3 memory systems: The response indicates the sampled level of DQ or the value of read DQS gate. This value is used by the MC to determine how to adjust the delay value. • MC Evaluation mode for LPDDR2 memory systems: The response indicates the value of read DQS gate. This value is used by the MC to determine how to adjust the delay value. This signal is not used for data eye training for LPDDR2 memory systems. The width of the **dfi_rdlvl_resp** signal is defined as a DFI term. The width will generally be defined as a bit per memory data slice, or as the same width as the memory data bus. If the response width is the same as the memory data bus width, then the response for gate training should be sent on the lowest bit of each data slice. This signal is only applicable for MCs connecting to PHYs operating in MC Evaluation or PHY Evaluation modes for read leveling.

TABLE 12. *Training Interface Signals*

Signal	From	Width	Default	Description
dfi_wrlvl_req	PHY	DFI Write Leveling PHY IF Width	0x0	PHY write leveling request. This is an optional signal for the PHY; other sources may be used to initiate write leveling or the MC may initiate write leveling independently. The PHY may drive independent write leveling requests from each data slice; however the MC must write level all data slices based on a single assertion of the **dfi_wrlvl_req** signal. If the PHY asserts the **dfi_wrlvl_req** signal, the MC must acknowledge the request by asserting the **dfi_wrlvl_en** signal within $\mathbf{t_{wrlvl\_resp}}$ cycles, after which the PHY should de-assert the **dfi_wrlvl_req** signal. The PHY should not assert this signal during initialization or a frequency change operation. The MC is responsible for any write leveling required during these operations.
dfi_wrlvl_mode	PHY	2 bits	- [a]	Write leveling support. Defines responsibility over the write leveling operation. The MC is required to support all of these modes. • 'b00 = Write leveling is not supported by the PHY. • 'b01 = MC Evaluation mode. The MC will enable and disable the write leveling logic in the PHY, analyze the results and adjust the delays. • 'b10 = PHY Evaluation mode. The MC enables and disables the write leveling logic in the PHY. The PHY contains logic to evaluate the results and set new delay values. • 'b11 = PHY Independent mode. The PHY performs all write leveling operations. This signal is required for all systems that support write leveling.
dfi_wrlvl_en	MC	DFI Write Leveling MC IF Width	0x0	PHY write leveling logic enable. This signal is asserted during write leveling. If the PHY initiated the write leveling request (**dfi_wrlvl_req**), then this serves as an acknowledge of that request. • 'b0 = Normal operation • 'b1 = Write leveling enabled. The assertion of this signal immediately triggers write leveling. This signal is only applicable for MCs connecting to PHYs operating in MC Evaluation or PHY Evaluation modes for write leveling.
dfi_wrlvl_cs_n	MC	DFI Chip Select Width	- [b]	Write leveling chip select. Indicates which chip select is currently active for write leveling. This signal is only applicable for MCs connecting to PHYs operating in MC Evaluation or PHY Evaluation modes for write leveling.

TABLE 12. *Training Interface Signals*

Signal	From	Width	Default	Description
dfi_wrlvl_delay_X	MC	DFI Write Leveling Delay Width	- [b]	Write leveling data delay. Indicates the programming of the PHY delay for the write DQS. The width of the **dfi_wrlvl_delay_X** signals is defined as a DFI term. In general, each memory data slice will be uniquely leveled and therefore the MC should provide a separate **dfi_wrlvl_delay_X** signal for each memory data slice X where **dfi_wrlvl_delay_0** corresponds to the first data slice. The width of each **dfi_wrlvl_delay_X** signal is defined by the programmability of the delay line. In some applications, the PHY may only use a subset of the delay signals provided by the MC. This signal is only applicable for MCs connecting to PHYs operating in MC Evaluation mode for write leveling.
dfi_wrlvl_load	MC	DFI Write Leveling MC IF Width	0x0	Write leveling load. The MC must send a 1 cycle pulse on this signal when it has updated any of the delay times (**dfi_wrlvl_delay_X**) for the next write leveling command. This signal is only applicable for MCs connecting to PHYs operating in MC Evaluation mode for write leveling.
dfi_wrlvl_strobe	MC	DFI Write Leveling MC IF Width	0x0	Write leveling strobe. Triggers the PHY write leveling strobe. This signal is only applicable for MCs connecting to PHYs operating in MC Evaluation or PHY Evaluation modes for write leveling.
dfi_wrlvl_resp	PHY	DFI Write Leveling Response Width	- [b]	Write leveling response. Response definition depends on the mode of operation: • PHY Evaluation mode: The response indicates that the PHY has completed write leveling and aligned the DQS relative to the memory clock. • MC Evaluation mode: The response indicates the sampled level of DQ. This value is used by the MC to determine how to adjust the delay value. The width of the **dfi_wrlvl_resp** signal is defined as a DFI term. The width will generally be defined as a bit per memory data slice, or as the same width as the data bus. This signal is only applicable for MCs connecting to PHYs operating in MC Evaluation or PHY Evaluation modes for write leveling.

a. The default value is defined by the PHY implementation.

b. This signal is not meaningful during initialization. Therefore, no default value is required to be maintained for this signal.

Timing parameters are relevant for certain Read and Write leveling modes and are identified accordingly in Table 13, “Training Interface Timing Parameters”. All timing parameters are defined only once for the interface and must apply to all PHY memory data slices.

TABLE 13. *Training Interface Timing Parameters*

Parameter	Defined By	Min	Max	Unit	Description
t_{rdlvl_dll}	PHY	1	- [a]	DFI clock cycles	Read leveling DLL delay. Specifies the minimum number of DFI clock cycles from when the MC asserts the **dfi_rdlvl_load** signal and updates the DLL delay in the appropriate **dfi_rdlvl_delay_X** or **dfi_rdlvl_gate_delay_X** signal to when the PHY is ready for the next read (DDR3) or mode register read (LPDDR2) command. This timing parameter is only applicable for MCs connecting to PHYs operating in MC Evaluation mode for read leveling.
t_{rdlvl_en}	MC	1	- [a]	DFI clock cycles	Read leveling enable time. For MC Evaluation mode, this specifies the minimum number of DFI clock cycles from the assertion of the **dfi_rdlvl_en** or **dfi_rdlvl_gate_en** signal to the first **dfi_rdlvl_load** signal assertion. For PHY Evaluation mode, this specifies the minimum number of DFI clock cycles from the assertion of the **dfi_rdlvl_en** or **dfi_rdlvl_gate_en** signal to the first read (DDR3) or mode register read (LPDDR2) command. This timing parameter is only applicable for MCs connecting to PHYs operating in MC Evaluation or PHY Evaluation modes for read leveling.
t_{rdlvl_load}	MC	1	- [a]	DFI clock cycles	Read leveling delay settling time. Specifies the minimum number of DFI clock cycles from when the delays are loaded on the **dfi_rdlvl_delay_X** or **dfi_rdlvl_gate_delay_X** signals to when the **dfi_rdlvl_load** signal may be asserted. This timing parameter is only applicable for MCs connecting to PHYs operating in MC Evaluation mode for read leveling.
t_{rdlvl_max}	MC	- [b]	- [a]	DFI clock cycles	Read leveling maximum time. Specifies the maximum number of DFI clock cycles that the MC will wait for a response (**dfi_rdlvl_resp**) to a read leveling enable signal (**dfi_rdlvl_en** or **dfi_rdlvl_gate_en**). This timing parameter is only applicable for MCs connecting to PHYs operating in PHY Evaluation mode for read leveling.
t_{rdlvl_resp}	MC	1	- [a]	DFI clock cycles	Read leveling response. Specifies the maximum number of DFI clock cycles after a read leveling request is asserted (**dfi_rdlvl_req** or **dfi_rdlvl_gate_req**) to when the MC will respond with a read leveling enable signal (**dfi_rdlvl_en** or **dfi_rdlvl_gate_en**).
$t_{rdlvl\_resplat}$	PHY	1	- [a]	DFI clock cycles	Read leveling response latency. Specifies the maximum number of DFI clock cycles from the assertion of a read (DDR3) or mode register read (LPDDR2) command to the guaranteed validity of the **dfi_rdlvl_resp** signal. This timing parameter is only applicable for MCs connecting to PHYs operating in MC Evaluation mode for read leveling.

TABLE 13. *Training Interface Timing Parameters*

Parameter	Defined By	Min	Max	Unit	Description
t_{rdlvl_rr}	PHY	- [b]	- [a]	DFI clock cycles	Read leveling command-to-command delay. For DDR3 memory systems, specifies the minimum number of DFI clock cycles after the assertion of a read command to the next read command. For LPDDR2 memory systems, specifies the minimum number of DFI clock cycles after the assertion of a mode register read command to the next mode register read command. This timing parameter is only applicable for MCs connecting to PHYs operating in MC Evaluation or PHY Evaluation modes for read leveling.
t_{wrlvl_dll}	PHY	1	- [a]	DFI clock cycles	Write leveling DLL delay. Specifies the minimum number of DFI clock cycles from when the MC asserts the **dfi_wrlvl_load** signal and updates the DLL delay in the appropriate **dfi_wrlvl_delay_X** signal to when the PHY is ready for the next **dfi_wrlvl_strobe** signal assertion. This timing parameter is only applicable for MCs connecting to PHYs operating in MC Evaluation mode for write leveling.
t_{wrlvl_en}	MC	1	- [a]	DFI clock cycles	Write leveling enable time. For MC Evaluation mode, this specifies the minimum number of DFI clock cycles from the assertion of the **dfi_wrlvl_en** signal to the first **dfi_wrlvl_load** signal assertion. For PHY Evaluation mode, this specifies the minimum number of DFI clock cycles from the assertion of the **dfi_wrlvl_en** signal to the first **dfi_wrlvl_strobe** signal assertion. This timing parameter is only applicable for MCs connecting to PHYs operating in MC Evaluation or PHY Evaluation modes for write leveling.
t_{wrlvl_load}	MC	1	- [a]	DFI clock cycles	Write leveling delay settling time. Specifies the minimum number of DFI clock cycles from when the delays are loaded on the **dfi_wrlvl_delay_X** signals to when the **dfi_wrlvl_load** signal may be asserted. This timing parameter is only applicable for MCs connecting to PHYs operating in MC Evaluation mode for write leveling.
t_{wrlvl_max}	MC	- [b]	- [a]	DFI clock cycles	Write leveling maximum time. Specifies the maximum number of DFI clock cycles that the MC will wait for a response (**dfi_wrlvl_resp**) to a write leveling enable signal (**dfi_wrlvl_en**). This timing parameter is only applicable for MCs connecting to PHYs operating in PHY Evaluation mode for write leveling.
t_{wrlvl_resp}	MC	1	- [a]	DFI clock cycles	Write leveling response. Specifies the maximum number of DFI clock cycles after a write leveling request is asserted (**dfi_wrlvl_req**) to when the MC will respond with a write leveling enable signal (**dfi_wrlvl_en**).

TABLE 13. *Training Interface Timing Parameters*

Parameter	Defined By	Min	Max	Unit	Description
$t_{wrlvl\_resplat}$	PHY	1	- [a]	DFI clock cycles	Write leveling response latency. Specifies the maximum number of DFI clock cycles from the assertion of **dfi_wrlvl_strobe** signal to the guaranteed validity of the **dfi_wrlvl_resp**. This timing parameter is only applicable for MCs connecting to PHYs operating in MC Evaluation mode for write leveling.
t_{wrlvl_ww}	PHY	- [b]	- [a]	DFI clock cycles	Write leveling write-to-write delay. Specifies the minimum number of DFI clock cycles after the assertion of the **dfi_wrlvl_strobe** signal to the next **dfi_wrlvl_strobe** signal assertion. This timing parameter is only applicable for MCs connecting to PHYs operating in MC Evaluation or PHY Evaluation modes for write leveling.

a. The DFI does not specify a maximum value. The range of values supported is an implementation-specific design parameter.

b. The DFI does not specify a minimum value. The range of values supported is an implementation-specific design parameter.

3.7 Low Power Control Interface

In a DDR memory subsystem, it may be advantageous to place the PHY in a low power state when the MC has knowledge that the memory subsystem will remain idle for a period of time. Depending on the state of the system, the MC will communicate state information to the PHY allowing the PHY to enter the appropriate power saving state. This optional interface consists of signals that are used to inform the PHY of a low power mode opportunity, as well as how quickly the MC will require the PHY to resume normal operation. More information on the low power control interface is provided in Section 4.11, “Low Power Control Handshaking”.

The DFI specification defines wakeup time as a specific number of cycles in which the PHY is required to respond to a signal change (the de-assertion of the **dfi_lp_req** signal) on the DFI. When the MC detects an idle time, it asserts the **dfi_lp_req** signal to the PHY and the **dfi_lp_wakeup** signal with the wakeup time required. The PHY can acknowledge the request or ignore it.

If the request is acknowledged by asserting the **dfi_lp_ack** signal, the protocol described in Section 4.11, “Low Power Control Handshaking” must be followed. The PHY may go into low power mode based on the wakeup time required, or it may choose to not change power states. If a low power mode is entered, the PHY should remain in low power mode as long as the request and acknowledge are both asserted. The PHY is not required to change power consumption even if it acknowledges the low power opportunity request from the memory controller. During low power handshaking, the DFI clock must maintain a valid and constant clock operating frequency.

The DFI specifies up to 16 different wakeup times, the time in which the PHY is required to exit a particular power mode (t_{lp_wakeup} cycles). Neither the MC nor the

PHY are required to support all of the wakeup times defined in this specification. Generally, the PHY should enter the lowest power state that it supports which allows low power exit within the required wakeup time. The signals and timing parameters for the low power control interface are listed in Table 14 and Table 15.

TABLE 14. *Low Power Control Interface Signals*

Signal	From	Width	Default	Description
dfi_lp_ack	PHY	1 bit	0x0	Low power acknowledge. The **dfi_lp_ack** signal is asserted to acknowledge the MC low power opportunity request. The PHY is not required to acknowledge this request. If the PHY chooses to acknowledge the request, the **dfi_lp_ack** signal must be asserted within $\mathbf{t_{lp\_resp}}$ cycles after the **dfi_lp_req** signal assertion. Once asserted, this signal should remain asserted until the **dfi_lp_req** signal de-asserts. The signal must de-assert within $\mathbf{t_{lp\_wakeup}}$ cycles after the **dfi_lp_req** signal de-asserts, indicating that the PHY is able to resume normal operation. If the PHY chooses to ignore the request, the **dfi_lp_ack** signal must remain de-asserted for the remainder of the low power mode opportunity. The **dfi_lp_req** signal will be asserted for at least $\mathbf{t_{lp\_resp}}$ cycles.

TABLE 14. *Low Power Control Interface Signals*

Signal	From	Width	Default	Description
dfi_lp_wakeup	MC	4 bits	- [a]	Low power wakeup time. The **dfi_lp_wakeup** signal indicates which one of the 16 wakeup times the MC is requesting for the PHY. The signal is only valid when the **dfi_lp_req** signal is asserted. The **dfi_lp_wakeup** signal must remain constant until the **dfi_lp_ack** signal is asserted. Once the request has been acknowledged, the MC may increase the **dfi_lp_wakeup** signal, permitting the PHY to enter a lower power state. The PHY is not required to change power states in response to the wakeup time change. The MC may not decrease this value once the request has been acknowledged. The value of the **dfi_lp_wakeup** signal at the time that the **dfi_lp_req** signal is de-asserted will set the $\mathbf{t_{lp\_wakeup}}$ time. • 'b0000 = $\mathbf{t_{lp\_wakeup}}$ is 16 cycles • 'b0001 = $\mathbf{t_{lp\_wakeup}}$ is 32 cycles • 'b0010 = $\mathbf{t_{lp\_wakeup}}$ is 64 cycles • 'b0011 = $\mathbf{t_{lp\_wakeup}}$ is 128 cycles • 'b0100 = $\mathbf{t_{lp\_wakeup}}$ is 256 cycles • 'b0101 = $\mathbf{t_{lp\_wakeup}}$ is 512 cycles • 'b0110 = $\mathbf{t_{lp\_wakeup}}$ is 1024 cycles • 'b0111 = $\mathbf{t_{lp\_wakeup}}$ is 2048 cycles • 'b1000 = $\mathbf{t_{lp\_wakeup}}$ is 4096 cycles • 'b1001 = $\mathbf{t_{lp\_wakeup}}$ is 8192 cycles • 'b1010 = $\mathbf{t_{lp\_wakeup}}$ is 16384 cycles • 'b1011 = $\mathbf{t_{lp\_wakeup}}$ is 32768 cycles • 'b1100 = $\mathbf{t_{lp\_wakeup}}$ is 65536 cycles • 'b1101 = $\mathbf{t_{lp\_wakeup}}$ is 131072 cycles • 'b1110 = $\mathbf{t_{lp\_wakeup}}$ is 262144 cycles • 'b1111 = $\mathbf{t_{lp\_wakeup}}$ is unlimited

TABLE 14. *Low Power Control Interface Signals*

Signal	From	Width	Default	Description
dfi_lp_req	MC	1 bit	0x0	Low power opportunity request. The **dfi_lp_req** signal is used by the MC to inform the PHY of an opportunity to switch to a low power mode. The MC must assert a constant value on the **dfi_lp_wakeup** signal while this signal is asserted before the request is acknowledged by the PHY through the assertion of the **dfi_lp_ack** signal or until $\mathbf{t_{lp\_resp}}$ cycles have elapsed. The MC may increase the value of the **dfi_lp_wakeup** signal if both the **dfi_lp_req** and **dfi_lp_ack** signals are asserted. Following the de-assertion of the **dfi_lp_req** signal, the PHY has $\mathbf{t_{lp\_wakeup}}$ cycles to resume normal operation and de-assert the **dfi_lp_ack** signal.

a. This signal is not meaningful during initialization. Therefore, no default value is required to be maintained for this signal.

TABLE 15. *Low Power Control Timing Parameters*

Parameter	Defined By	Min	Max	Unit	Description
$\mathbf{t_{lp\_resp}}$	MC	1[a]	7	DFI clock cycles	Specifies the maximum number of DFI clock cycles after the assertion of the **dfi_lp_req** signal to the assertion of the **dfi_lp_ack** signal.
$\mathbf{t_{lp\_wakeup}}$	MC	0	-[b]	DFI clock cycles	Specifies the maximum number of DFI clock cycles that the **dfi_lp_ack** signal may remain asserted after the de-assertion of the **dfi_lp_req** signal. The **dfi_lp_ack** signal may de-assert at any cycle after the de-assertion of the **dfi_lp_req** signal.

a. It is recommended to fix this timing parameter at 7 cycles.

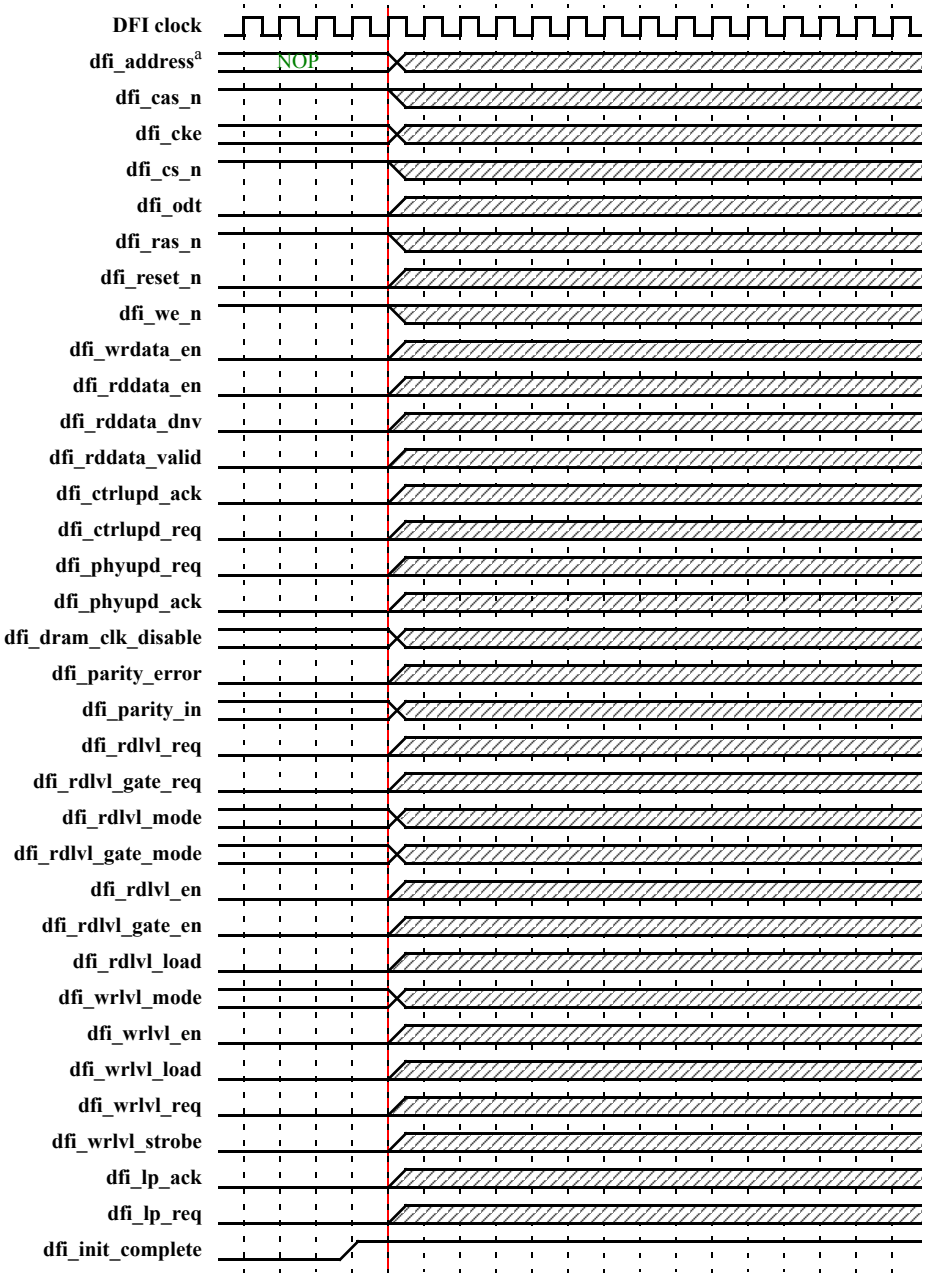
b. There is no maximum value defined for this timing parameter.

4.0 Functional Use

4.1 Initialization

The DFI specification requires that, as long as the **dfi_init_complete** signal is not asserted, the DFI signals that communicate commands or status must maintain their default value. The signals shown in Figure 2, “Dependency on dfi_init_complete” must wait for the **dfi_init_complete** signal to be asserted before they can be driven with a value other than the default value defined in the DFI specification.

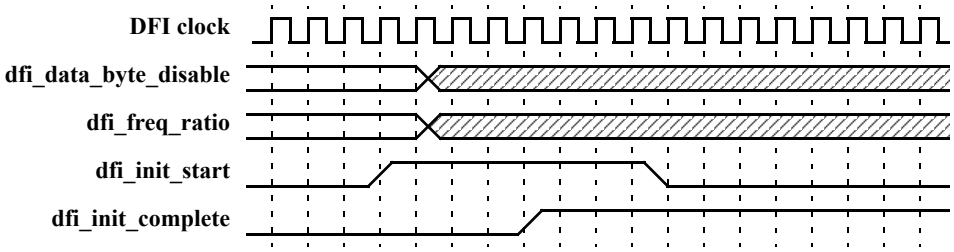
FIGURE 2. *Dependency on **dfi_init_complete***



a) The **dfi_address** signal does not have a default value in most cases. However, for an LPDDR2 memory system, the **dfi_address** bus must drive a NOP command until the **dfi_init_complete** signal is asserted.

Two signals of the status interface are used to convey setup information about the system to the PHY: **dfi_data_byte_disable** and **dfi_freq_ratio**. The **dfi_data_byte_disable** signal informs the PHY if the MC is disabling certain byte lanes from use during data transfers, and the **dfi_freq_ratio** signal identifies the MC:PHY frequency ratio. Once either or both of these signals have been set with valid values during initialization, the MC will drive the **dfi_init_start** signal to the PHY. The PHY may optionally use the **dfi_init_start** signal assertion to know that these status signals are valid. After the PHY drives the **dfi_init_complete** signal to the MC, the MC may release the **dfi_init_start** signal. This scenario is shown in Figure 3, "System Setting Signals". Note that the PHY is NOT required to wait for the **dfi_init_start** signal assertion prior to asserting **dfi_init_complete**; the **dfi_init_start** signal is only provided as an indicator of the status of the **dfi_data_byte_disable** and/or **dfi_freq_ratio** signals. When the MC has asserted the **dfi_init_start** signal, if the **dfi_init_complete** signal has already been asserted, the MC may immediately de-assert the **dfi_init_start** signal.

FIGURE 3. *System Setting Signals*



The DFI specification does not impose or dictate a reset sequence for either the PHY or the MC. However, the assertion of the **dfi_init_complete** signal signifies that the PHY is ready to respond to any assertions on the DFI by the MC. This does not ensure data integrity to the DRAMs, only that the PHY can respond to the changes with appropriate responses on the DFI. The PHY must guarantee the integrity of the address and control interface to the DRAMs prior to asserting the **dfi_init_complete** signal. Note that the DFI does not impose nor dictate any need for any type of signal training prior to DFI signal assertion.

For LPDDR2 memory systems, the **dfi_address** bus must drive a NOP command until the **dfi_init_complete** signal is asserted and the signals **dfi_bank**, **dfi_cas_n**, **dfi_ras_n** and **dfi_we_n** are unused and must remain at a constant value when the DFI bus is being used.

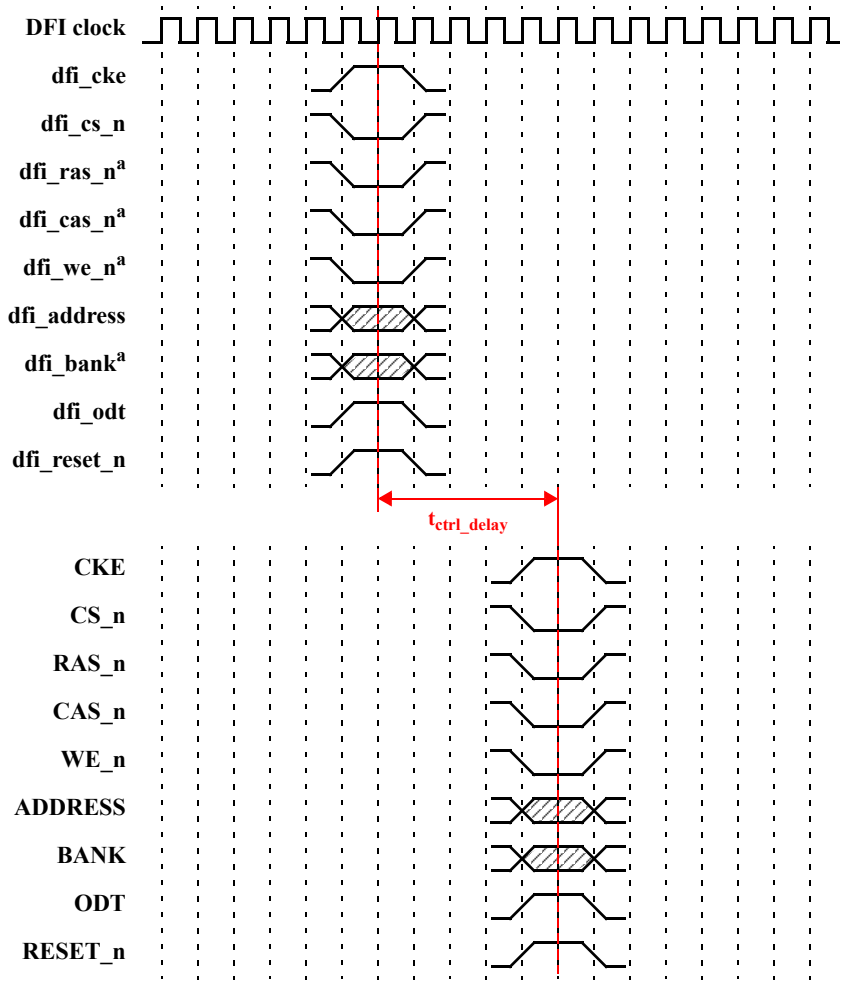
Some of the training interface signals must remain at default until after the assertion of the **dfi_init_complete** signal. No default value must be maintained for the following signals: **dfi_bank**, **dfi_wrdata**, **dfi_wrdata_mask**, **dfi_rddata**, **dfi_phyupd_type**, **dfi_rdlvl_cs_n**, **dfi_rdlvl_edge**, **dfi_rdlvl_delay_X**, **dfi_rdlvl_gate_delay_X**, **dfi_rdlvl_resp**, **dfi_wrlvl_cs_n**, **dfi_wrlvl_delay_X**, **dfi_wrlvl_resp** and **dfi_lp_wakeup**. The **dfi_address** signal also has no default value except for LPDDR2 memory systems.

4.2 Control Signals

The DFI control signals **dfi_address**, **dfi_bank**, **dfi_cas_n**, **dfi_cke**, **dfi_cs_n**, **dfi_reset_n**, **dfi_odt**, **dfi_ras_n** and **dfi_we_n** correlate to the DRAM control signals. For more information on these signals, refer to Section 3.1, “Control Interface”.

These control signals are expected to be driven to the memory devices. The DFI relationship of the control signals is expected to be maintained at the PHY-DRAM boundary; meaning that any delays should be consistent across all signals and is defined through the timing parameter $\mathbf{t_{ctrl\_delay}}$. Refer to Figure 4, “DFI Control Interface Signal Relationships” for a graphical representation.

FIGURE 4. *DFI Control Interface Signal Relationships*



a) For LPDDR2, these signals are not used and should be held in an idle state.

The system may not be using all of the pins on the DRAM interface such as additional banks, chip selects, etc.; However, these signals must still be driven through the DFI and may not be left floating.

4.3 Write Transactions

The write transaction interface of the DFI includes the write data (**dfi_wrdata**), write data mask (**dfi_wrdata_mask**), and write data enable (**dfi_wrdata_en**) signals as well as the $\mathbf{t_{phy\_wrlat}}$ and $\mathbf{t_{phy\_wrdata}}$ parameters. For more information on these signals, refer to Section 3.2, “Write Data Interface”.

The **dfi_wrdata_en** signal must be asserted $\mathbf{t_{phy\_wrlat}}$ cycles after the assertion of the corresponding write command on the DFI, and for the number of cycles required to complete the write data transfer sent on the DFI control interface. For contiguous write commands, the **dfi_wrdata_en** signal will be asserted $\mathbf{t_{phy\_wrlat}}$ cycles after the first write command of the stream and remain asserted for the entire length of the data stream.

The associated write data (**dfi_wrdata**) and masking (**dfi_wrdata_mask**) will be sent $\mathbf{t_{phy\_wrdata}}$ cycles after the assertion of the **dfi_wrdata_en** signal on the DFI. The **dfi_wrdata_en** signal must de-assert $\mathbf{t_{phy\_wrdata}}$ cycles before the last valid data is transferred on the **dfi_wrdata** bus.

Six situations are presented in Figure 5, Figure 6, Figure 7, Figure 8, Figure 9 and Figure 10. All six situations show system behavior with two write transactions.

Figure 5, shows back-to-back writes for a system with a $\mathbf{t_{phy\_wrlat}}$ of zero and a $\mathbf{t_{phy\_wrdata}}$ of one. The **dfi_wrdata_en** signal is asserted with the write command for this situation, and is asserted for two cycles per command to inform the DFI that two cycles of DFI data will be sent for each write command. The timing parameters and the timing of the write commands allow the **dfi_wrdata_en** signal and the **dfi_wrdata** stream to be sent contiguously.

FIGURE 5. *Back-to-Back Writes (DDR1 Example)*

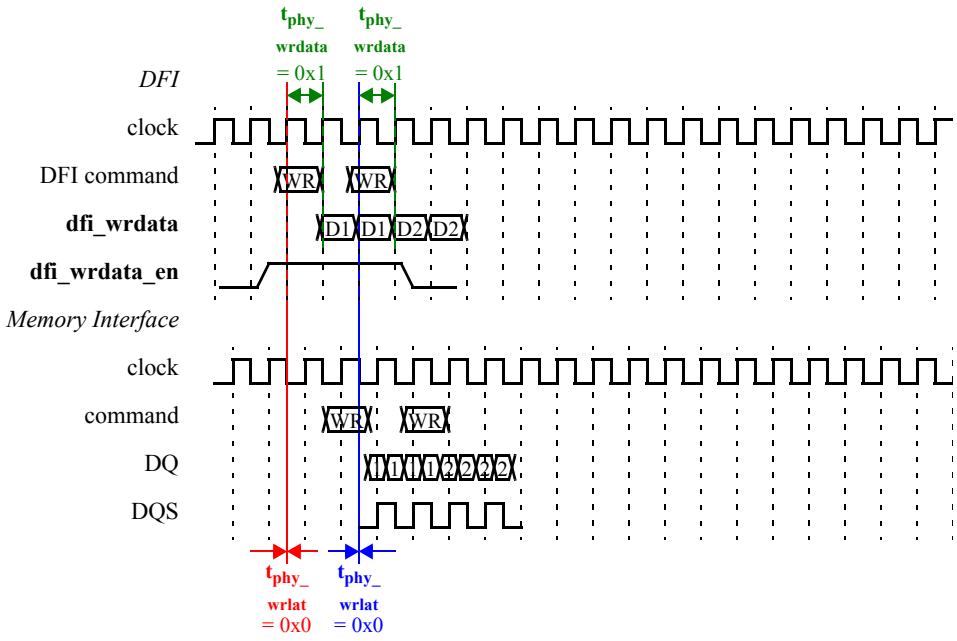


Figure 6 shows an interrupted write command. The **dfi_wrdata_en** signal should be asserted for 4 cycles for each of these write transactions. However, since the first write is interrupted, the **dfi_wrdata_en** signal is asserted for a portion of the first transaction and the complete second transaction. The **dfi_wrdata_en** signal will not de-assert between write commands, and the **dfi_wrdata** stream will be sent contiguously for a portion of the first command and the complete second command.

FIGURE 6. *Back-to-Back Interrupted Contiguous Writes (DDR2 Example)*

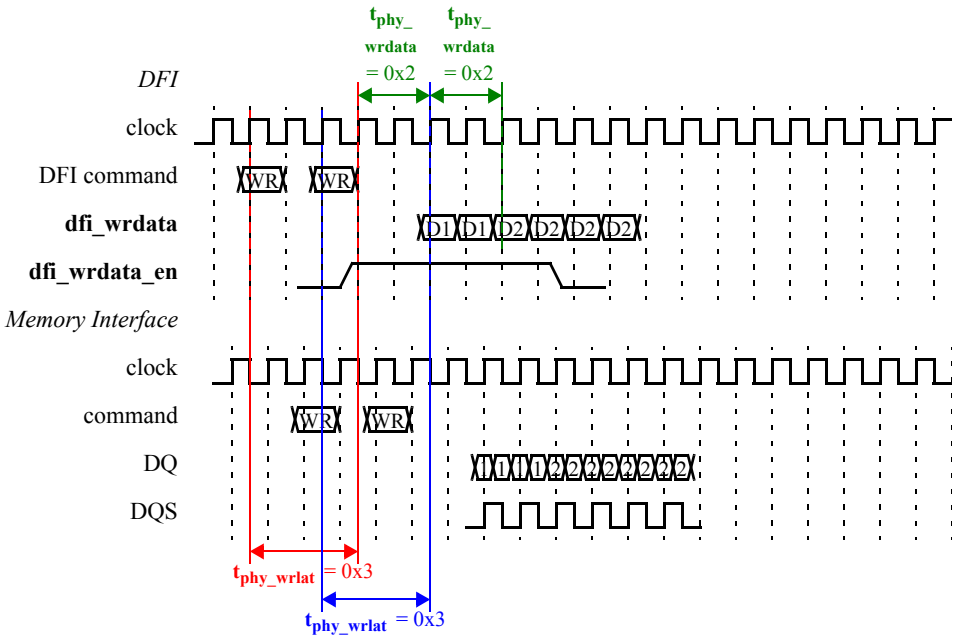


Figure 7 shows back-to-back burst-of-8 writes. The **dfi_wrdata_en** signal must be asserted for 4 cycles for each of these write transactions.

FIGURE 7. *Back-to-Back Writes (DDR3 Example)*

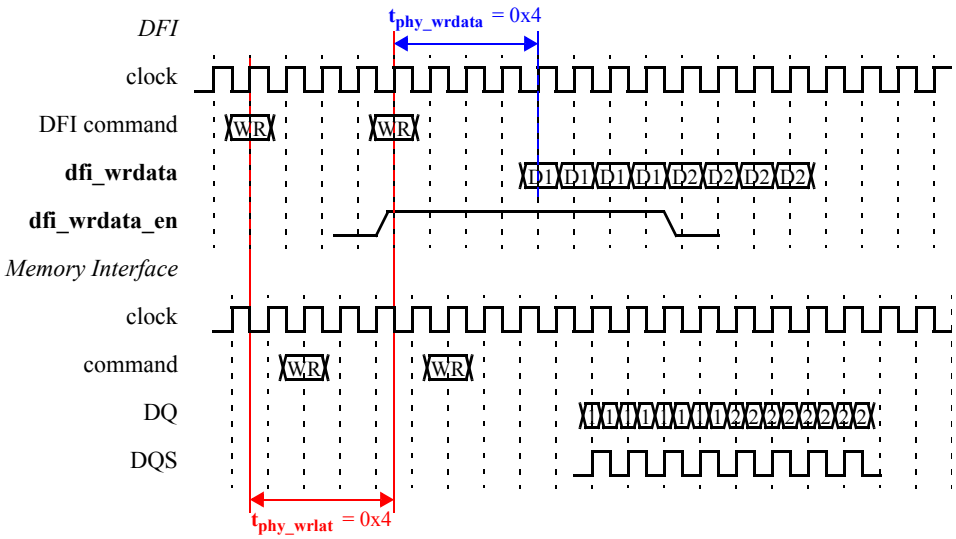


Figure 8, Figure 9, Figure 10 and Figure 11 also show two complete write commands, with different $\mathbf{t_{phy\_wrlat}}$ and $\mathbf{t_{phy\_wrdata}}$ timing parameters and for different memory types. The **dfi_wrdata_en** signal will be asserted for two cycles for each write transaction. The $\mathbf{t_{phy\_wrlat}}$ timing and the timing between the write commands causes the **dfi_wrdata_en** signal to be de-asserted between commands. As a result, the **dfi_wrdata** stream will be non-contiguous.

FIGURE 8. *Two Independent Writes (DDR1 Example)*

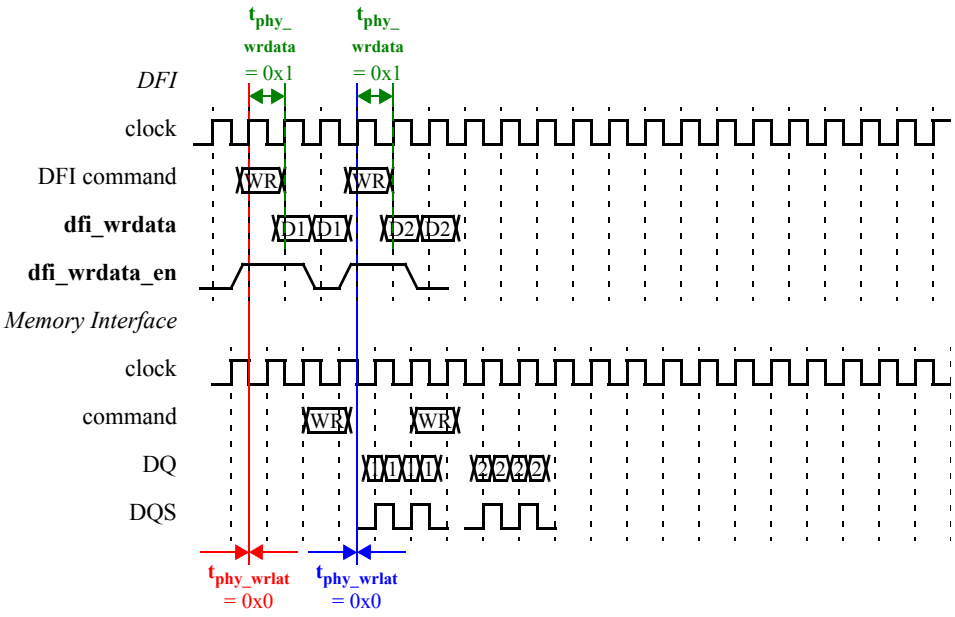


FIGURE 9. *Two Independent Writes (DDR2 Example)*

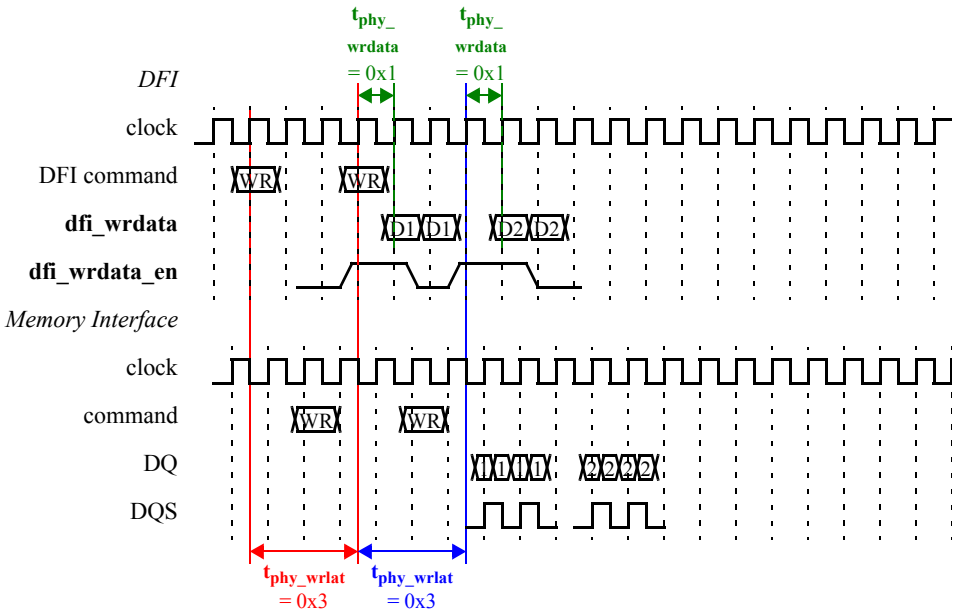


FIGURE 10. *Two Independent Writes (DDR3 Example)*

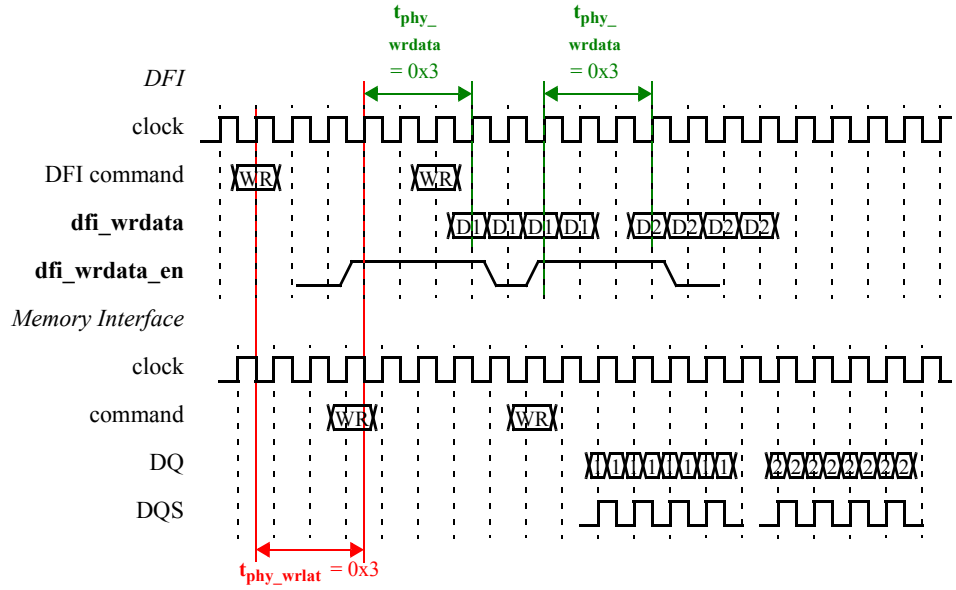
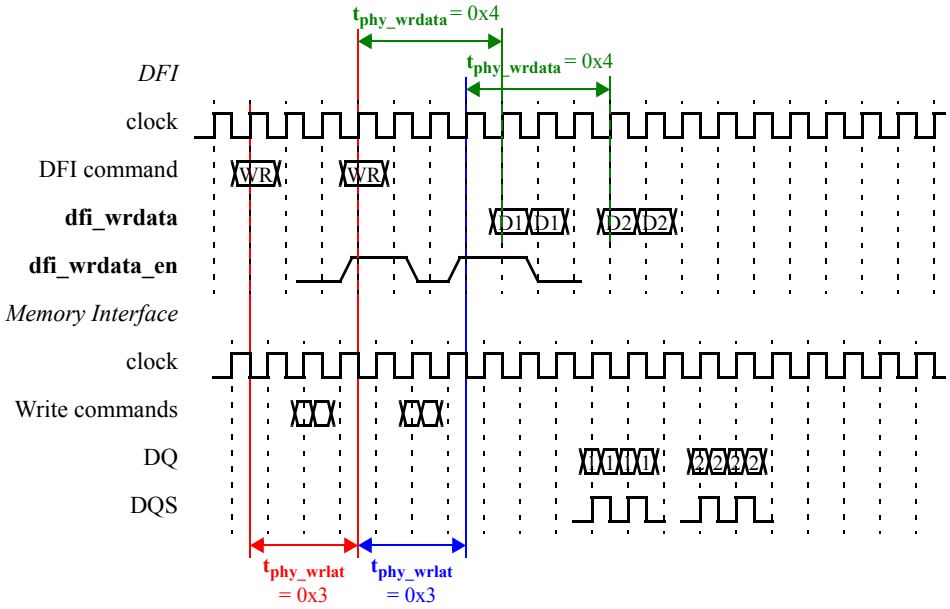


FIGURE 11. *Two Independent Writes (LPDDR2 Example)*



4.4 Read Transactions

The read transaction portion of the DFI is defined by the read data enable (**dfi_rddata_en**), read data (**dfi_rddata**), the read data not valid for LPDDR2 memory systems (**dfi_rddata_dnv**) and the valid (**dfi_rddata_valid**) signals as well as the $\mathbf{t_{rddata\_en}}$ and $\mathbf{t_{phy\_rdlat}}$ timing parameters. For more information on these signals, refer to Section 3.3, “Read Data Interface”.

For the DFI, the read data must be returned from the PHY within a maximum delay which is the sum of the $\mathbf{t_{rddata\_en}}$ and $\mathbf{t_{phy\_rdlat}}$ timing parameters. The $\mathbf{t_{rddata\_en}}$ is a fixed delay, but the $\mathbf{t_{phy\_rdlat}}$ is defined as a maximum value. The delay can be adjusted as long as both the MC and the PHY coordinate the change such that the DFI specification is still maintained. Both parameters may be expressed as equations based on other fixed system parameters.

The **dfi_rddata_en** signal must be asserted $\mathbf{t_{rddata\_en}}$ cycles after the assertion of the corresponding read command on the DFI, and the **dfi_rddata_en** signal must be asserted for the number of cycles of read data that the DFI is expecting. For contiguous read commands, the **dfi_rddata_en** signal will be asserted $\mathbf{t_{rddata\_en}}$ cycles after the first read command of the stream and remain asserted for the entire length of the data stream. The data will be returned, with the **dfi_rddata_valid** signal asserted, within $\mathbf{t_{phy\_rdlat}}$ cycles after the **dfi_rddata_en** signal for that command is asserted. For LPDDR2 memory systems, the **dfi_rddata_dnv** signal has the same timing as the **dfi_rddata** signal.

Seven situations are presented in Figure 12, Figure 13, Figure 14, Figure 15, Figure 16, Figure 18 and Figure 19.

Figure 12 shows a single read transaction. In this case, the **dfi_rddata_en** signal is asserted for two cycles to inform the DFI that two cycles of DFI data are expected and data is returned $\mathbf{t_{phy\_rdlat}}$ cycles after the **dfi_rddata_en** signal assertion.

FIGURE 12. *Single Read Transaction of 2 Data Words*

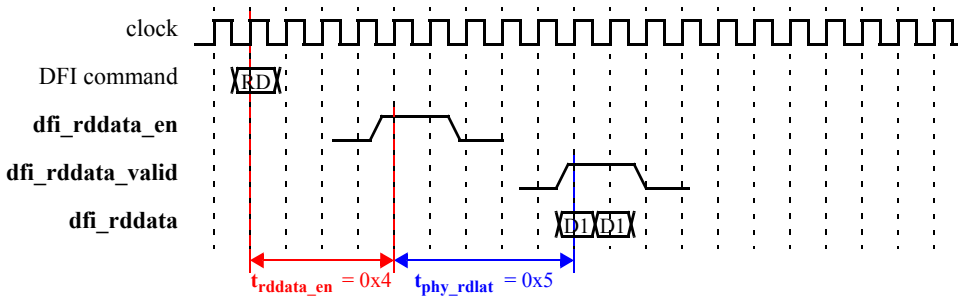


Figure 13 shows a single read transaction where the data is returned in less than the maximum delay. The data returns one cycle less than the maximum PHY read latency.

FIGURE 13. *Single Read Transaction of 4 Data Words*

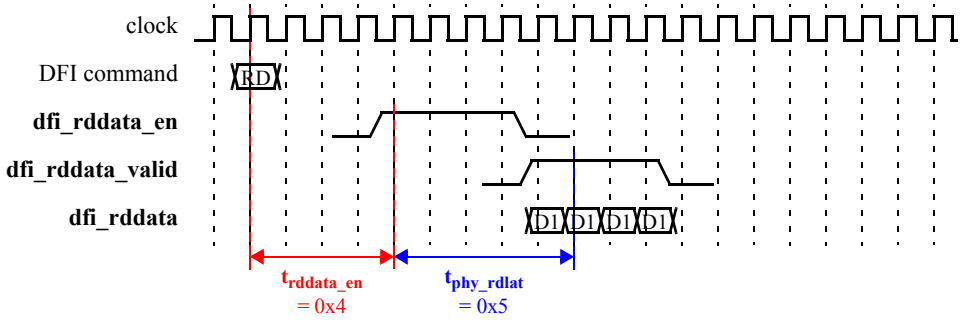


Figure 14 shows an interrupted read command. The **dfi_rddata_en** signal must be asserted for 4 cycles for each of these read transactions. However, since the first read is interrupted, the **dfi_rddata_en** signal is asserted for a portion of the first transaction and the complete second transaction. The **dfi_rddata_en** signal will not de-assert between read commands.

FIGURE 14. *Back-to-Back Read Transactions with First Read Burst Interrupted (DDR1 Example BL=8)*

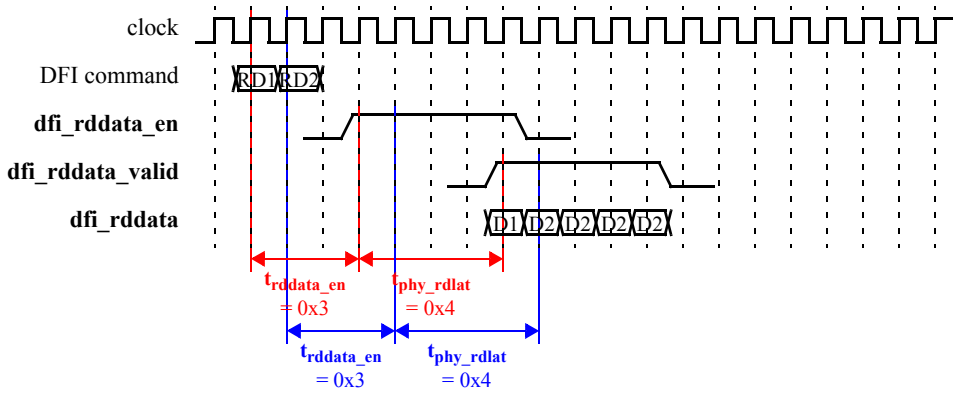
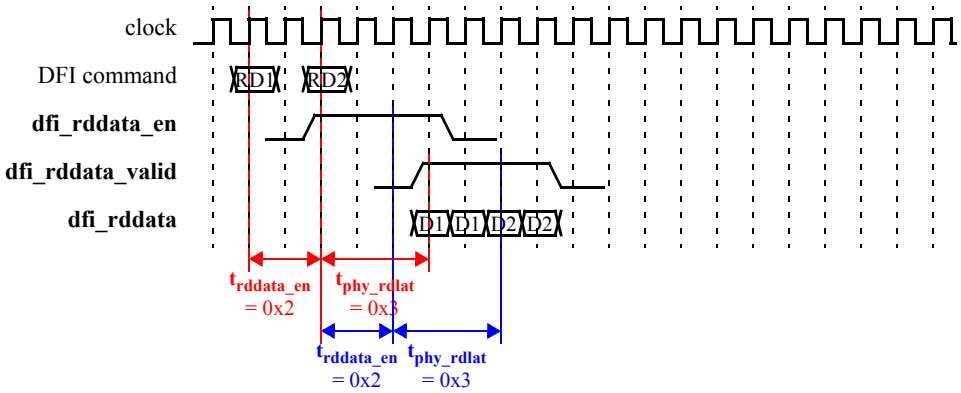


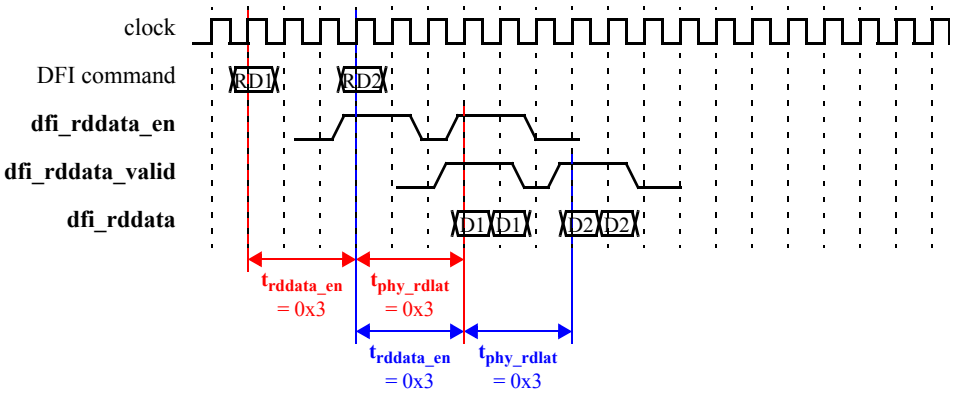
Figure 15 and Figure 16 also show two complete read transactions. The **dfi_rddata_en** signal will be asserted for two cycles for each read transaction. In Figure 15, the values for the timing parameters are such that the read data will be returned in a contiguous data stream for both transactions. Therefore, the **dfi_rddata_en** signal and the **dfi_rddata_valid** signal are each asserted for the complete read data stream.

FIGURE 15. *Two Independent Read Transactions (DDR1 Example)*



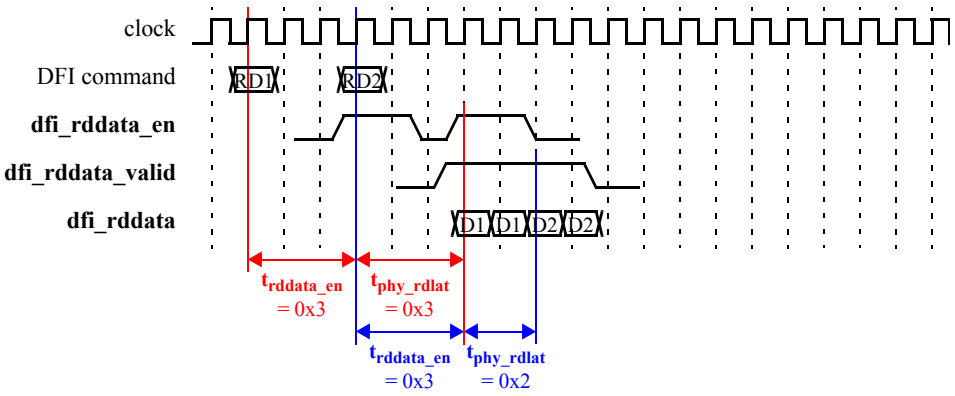
In Figure 16, the $\mathbf{t_{rddata\_en}}$ timing and the timing between the read commands causes the **dfi_rddata_en** signal to be de-asserted between commands. As a result, the **dfi_rddata_valid** signal will be de-asserted between commands and the **dfi_rddata** stream will be non-contiguous.

FIGURE 16. *Two Independent Read Transactions (DDR2 Example)*



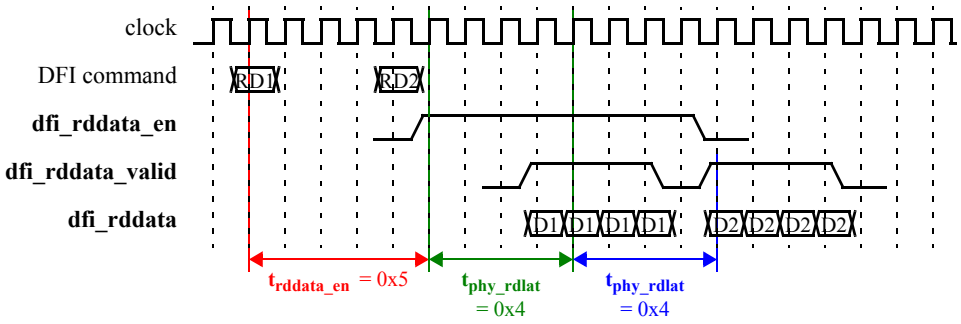
In Figure 17, the effective $\mathbf{t_{phy\_rdlat}}$ for the two transactions is different. This results in a situation in which the **dfi_rddata_valid** signal will remain asserted across commands and the **dfi_rddata** stream will be contiguous.

FIGURE 17. *Two Independent Read Transactions (DDR2 Example)*



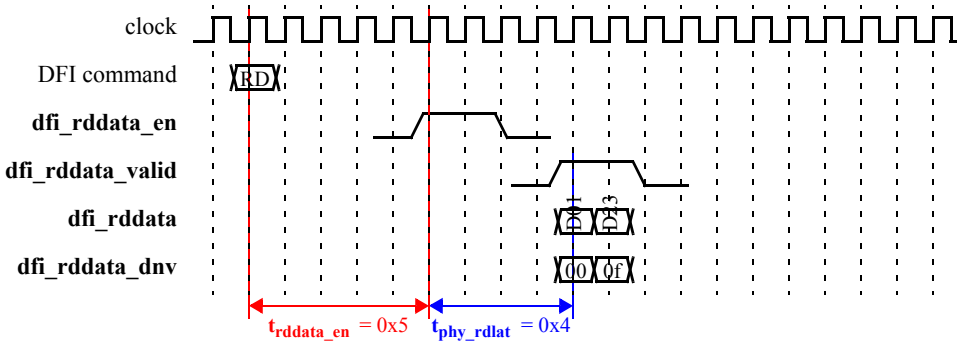
The data may return to the DFI in fewer cycles than maximum delay. In Figure 18, the first read data transfer is returned in three cycles, even though the $\mathbf{t_{phy\_rdlat}}$ timing parameter is set to four cycles. The second read data transfer is returned in the maximum of four cycles.

FIGURE 18. *Two Independent Read Transactions (DDR3 Example)*



LPDDR2 memories define a new transaction type of mode register read (MRR). From the DFI perspective, a mode register read is handled like any other read command and utilizing the same signals. Figure 19 shows a read transaction for a LPDDR2 memory device.

FIGURE 19. *Example Read Transactions with LPDDR2*



4.5 PHY Update

The DFI contains signals to support a MC-initiated and a PHY-initiated update process. The signals used in the update interface are: **dfi_ctrlupd_req**, **dfi_ctrlupd_ack**, **dfi_phyupd_req**, **dfi_phyupd_type** and **dfi_phyupd_ack**. For more information on these signals, refer to Section 3.4, “Update Interface”.

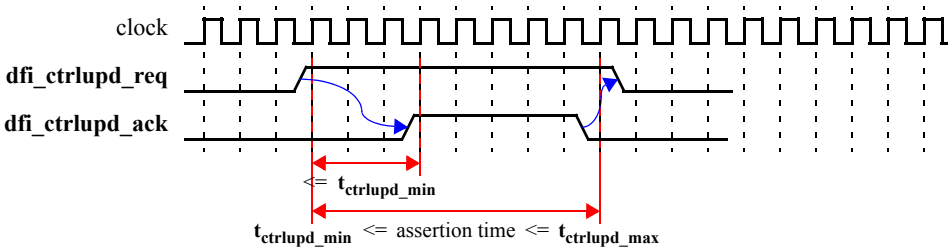
4.5.1 MC-Initiated Update

During normal operation, the MC may encounter idle time during which no commands are being issued to the memory devices and all outstanding read and write data have have been transferred on the DFI. Assertion of the **dfi_ctrlupd_req** signal indicates the control, read and write interfaces on the DFI are idle. While the **dfi_ctrlupd_ack** signal is asserted, the DFI bus may only be used for commands related to the update process.

The MC guarantees that **dfi_ctrlupd_req** signal will be asserted for at least $\mathbf{t_{ctrlupd\_min}}$ cycles, allowing the PHY time to respond. The PHY may respond or ignore the update request. To acknowledge the request, the **dfi_ctrlupd_ack** signal must be asserted while the **dfi_ctrlupd_req** signal is asserted. The **dfi_ctrlupd_ack** signal must de-assert at least one cycle before $\mathbf{t_{ctrlupd\_max}}$ expires.

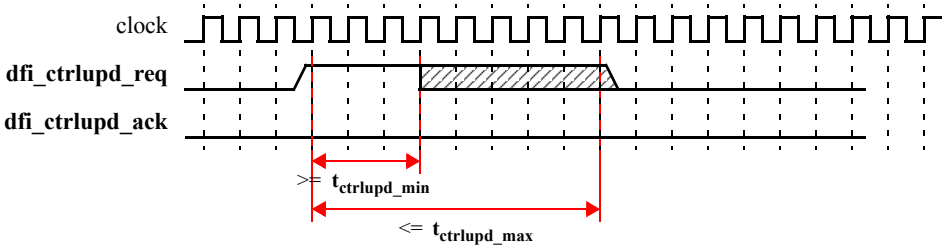
The MC must hold the **dfi_ctrlupd_req** signal as long as the **dfi_ctrlupd_ack** signal is asserted, and must de-assert the **dfi_ctrlupd_req** signal before $\mathbf{t_{ctrlupd\_max}}$ expires. Note that the number of cycles after the **dfi_ctrlupd_ack** signal de-asserts before the **dfi_ctrlupd_req** signal de-asserts is not specified by the DFI. This situation is shown in Figure 20.

FIGURE 20. *MC-Initiated Update Timing Diagram*



It is important to note that the **dfi_ctrlupd_ack** signal is not required to assert when the **dfi_ctrlupd_req** signal is asserted. The MC must assert the **dfi_ctrlupd_req** signal for at least $\mathbf{t_{ctrlupd\_min}}$ within every $\mathbf{t_{ctrlupd\_interval}}$ cycles, but the total number of cycles that the **dfi_ctrlupd_req** signal is asserted must not exceed $\mathbf{t_{ctrlupd\_max}}$. This scenario is shown in Figure 21.

FIGURE 21. *MC-Initiated Update with No Response*



4.5.2 PHY-Initiated Update

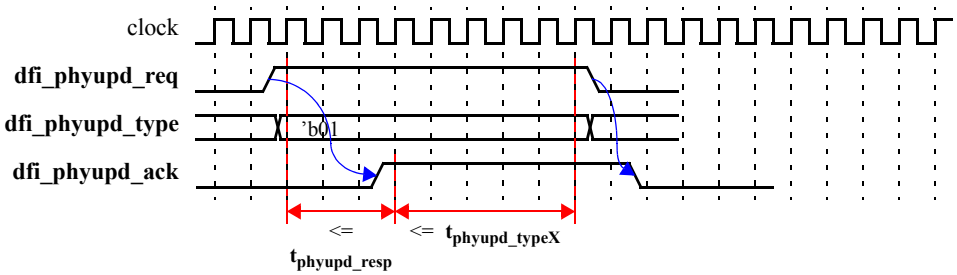
The PHY may also trigger the DFI into an idle state. This update process utilizes three signals: **dfi_phyupd_req**, **dfi_phyupd_type** and **dfi_phyupd_ack**. The **dfi_phyupd_req** signal indicates the need for idle time on the DFI, the **dfi_phyupd_type** signal defines the type of update required, and the **dfi_phyupd_ack** signal is the MC's response signal. Four update types are specified by the DFI.

To request an update, the **dfi_phyupd_type** signal must be valid when the **dfi_phyupd_req** signal is asserted. The $\mathbf{t_{phyupd\_typeX}}$ parameters indicate the number of cycles of idle time on the DFI control, read and write data interfaces being requested. The **dfi_phyupd_ack** signal must assert within $\mathbf{t_{phyupd\_resp}}$ cycles after the assertion of the **dfi_phyupd_req** signal.

When the **dfi_phyupd_ack** signal is asserted, it must remain asserted until the **dfi_phyupd_req** signal de-asserts or until $\mathbf{t_{phyupd\_typeX}}$ cycles have expired. The **dfi_phyupd_ack** signal must de-assert one cycle after the de-assertion of the **dfi_phyupd_req** signal. While the **dfi_phyupd_ack** signal is asserted, the DFI bus may only be used for commands related to the update process.

Unlike MC-initiated updates, the MC must respond to a PHY update request as shown in Figure 22.

FIGURE 22. *PHY-Initiated Update Timing Diagram*



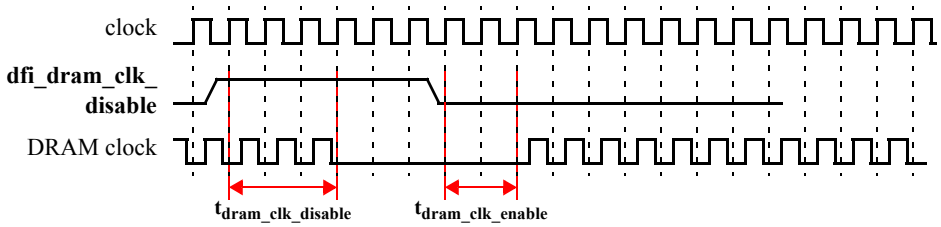
4.6 DFI Clock Disabling

The DFI contains a **dfi_dram_clk_disable** signal which controls the DRAM clock signal to the DRAM device(s). In the default state, the DRAM clock functions normally and the **dfi_dram_clk_disable** bits are all de-asserted. If the system requires the clocks of the memory device(s) to be disabled, then the **dfi_dram_clk_disable** signal will be asserted. For more information on the **dfi_dram_clk_disable** signal, refer to Section 3.5, "Status Interface".

Two timing parameters $\mathbf{t_{dram\_clk\_disable}}$ and $\mathbf{t_{dram\_clk\_enable}}$ indicate the number of DFI cycles that the PHY requires to respond to the assertion and de-assertion of the **dfi_dram_clk_disable** signal. The $\mathbf{t_{dram\_clk\_disable}}$ value determines the number of

DFI cycles in which a rising edge of the **dfi_dram_clk_disable** signal affects the DRAM clock and $\mathbf{t_{dram\_clk\_enable}}$ sets the number of cycles required for the DRAM clock to be active again, as shown in Figure 23.

FIGURE 23. *DRAM Clock Disable Behavior*



4.7 Frequency Ratios Across the DFI

In a DDR memory subsystem, it may be advantageous to operate the PHY at a higher frequency than the MC. If the PHY operates at a multiple of the MC frequency, the PHY transfers data at a higher data rate relative to the DFI clock and the MC has the option to execute multiple commands in a single DFI clock cycle. The DFI is defined at the MC to PHY boundary and therefore operates in the clock frequency domain of the MC.

The DFI specification supports a 1:2 or 1:4 MC to PHY frequency ratio, defining the relationship of the reference clocks for the MC and the PHY. The DFI PHY clock will always be the same frequency as the memory clock, which is 1/2 the DDR data rate for the memory. The MC clock is always the DFI clock and all DFI signals are referenced from this clock.

DFI signals may be sent or received on the PHY clock, provided that they reference the rising edge of the DFI clock and the clock is phase aligned. The MC communicates frequency ratio settings to the PHY on the **dfi_freq_ratio** signal. This signal is only required for devices using this frequency ratio protocol.

4.7.1 Frequency Ratio Clock Definition

The DFI clock and the DFI PHY clock must be phase-aligned and at a 1:2 or 1:4 frequency ratio relative to one another. Some DFI signals from the MC to the PHY must communicate information about the signal in reference to the DFI PHY clock to maintain the correct timing information. Therefore, the DFI PHY clock is described in terms of phases, where the number of clock phases for a system is the ratio of the DFI PHY clock to the DFI clock. Refer to Figure 24 and Figure 25 for the clock phase definitions for the 2 types of systems.

FIGURE 24. *Frequency Ratio 1:2 Phase Definition*

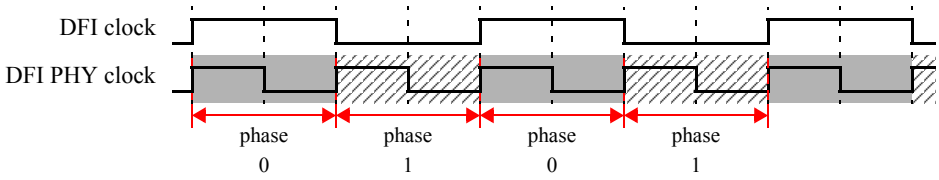
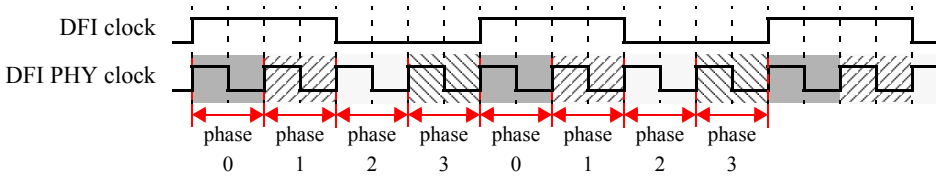


FIGURE 25. *Frequency Ratio 1:4 Phase Definition*



4.7.2 Interface Signals with Frequency Ratio Systems

The DFI specification supports the ability to send a unique command on each phase of the DFI PHY clock. To communicate this information to the PHY, the DFI specification defines commands for a frequency ratio system in a vectored format. The PHY must maintain this information to preserve the timing relationships between commands and data. Therefore, for frequency ratio systems, the control signal interface, the write data interface and the read data enable signal will all be suffixed with a "_pN" where N is the phase number. As an example, for a 1:2 frequency ratio system, instead of a single **dfi_address** signal, there will be 2 signals: **dfi_address_p0** and **dfi_address_p1**. The read data signal, read data valid and read data not valid signals will be suffixed with a "_wN" where N is the DFI data word. More information on the read data interface for frequency ratio systems is provided in Section 4.7.2.2, "Read Data Interface". The phase 0 or DFI data word 0 suffixes are not required.

There is flexibility in system setup for frequency ratio systems. The MC may be implemented to support command output on a single phase or on multiple phases. Even if multiple phases are supported, the MC is not required to implement or drive every phase of a signal. Only phases where a command will be sent must be implemented and driven. The exceptions to the rule are the **dfi_cke_pN** and **dfi_odt_pN** signals. These two signals are not necessarily driven in the same phase as the rest of the command. Therefore, these signals must be implemented for all phases of the clock to allow flexibility in timing.

The PHY must be able to accept a command on all phases to be DFI compliant. If the MC is only using certain phases, the PHY must be appropriately connected to properly interpret the command stream.

There is no requirement that signals must be implemented in the same way across the interfaces. For example, in a 2T implementation, the **dfi_cs_n_pN** signal may be driven by the MC on all clock phases, but the **dfi_ras_n_pN**, **dfi_cas_n_pN** and **dfi_we_n_pN** may only be driven by the MC on half of the phases.

Figure 26 demonstrates an example command stream for a 1:2 frequency ratio system and how the PHY in this system would interpret the DFI signals. In this example, a command is only sent on phase 0 and ODT information is provided on both phases. PHY timing is shown for illustrative purposes only.

FIGURE 26. *Example 1:2 Frequency Ratio Command Stream*

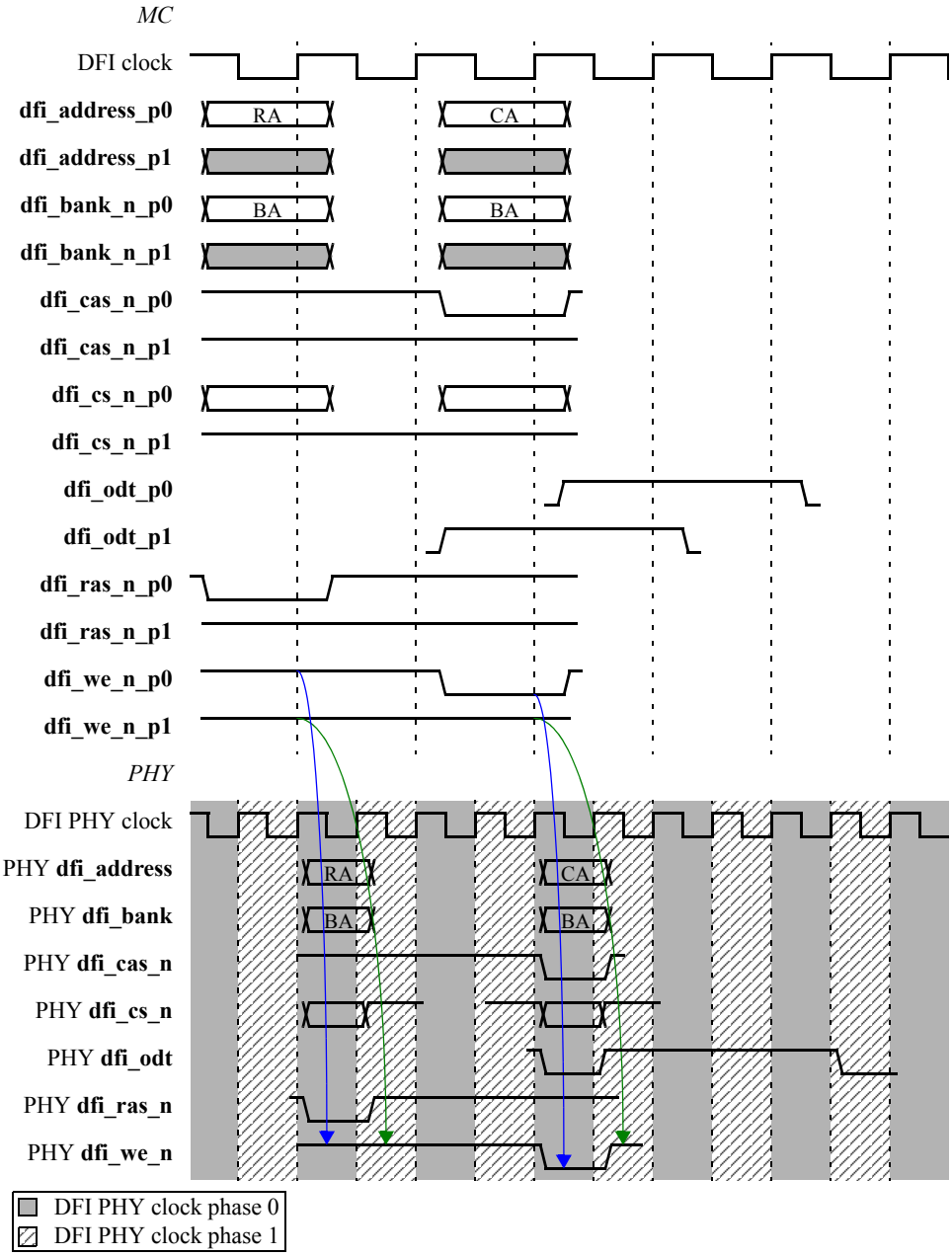
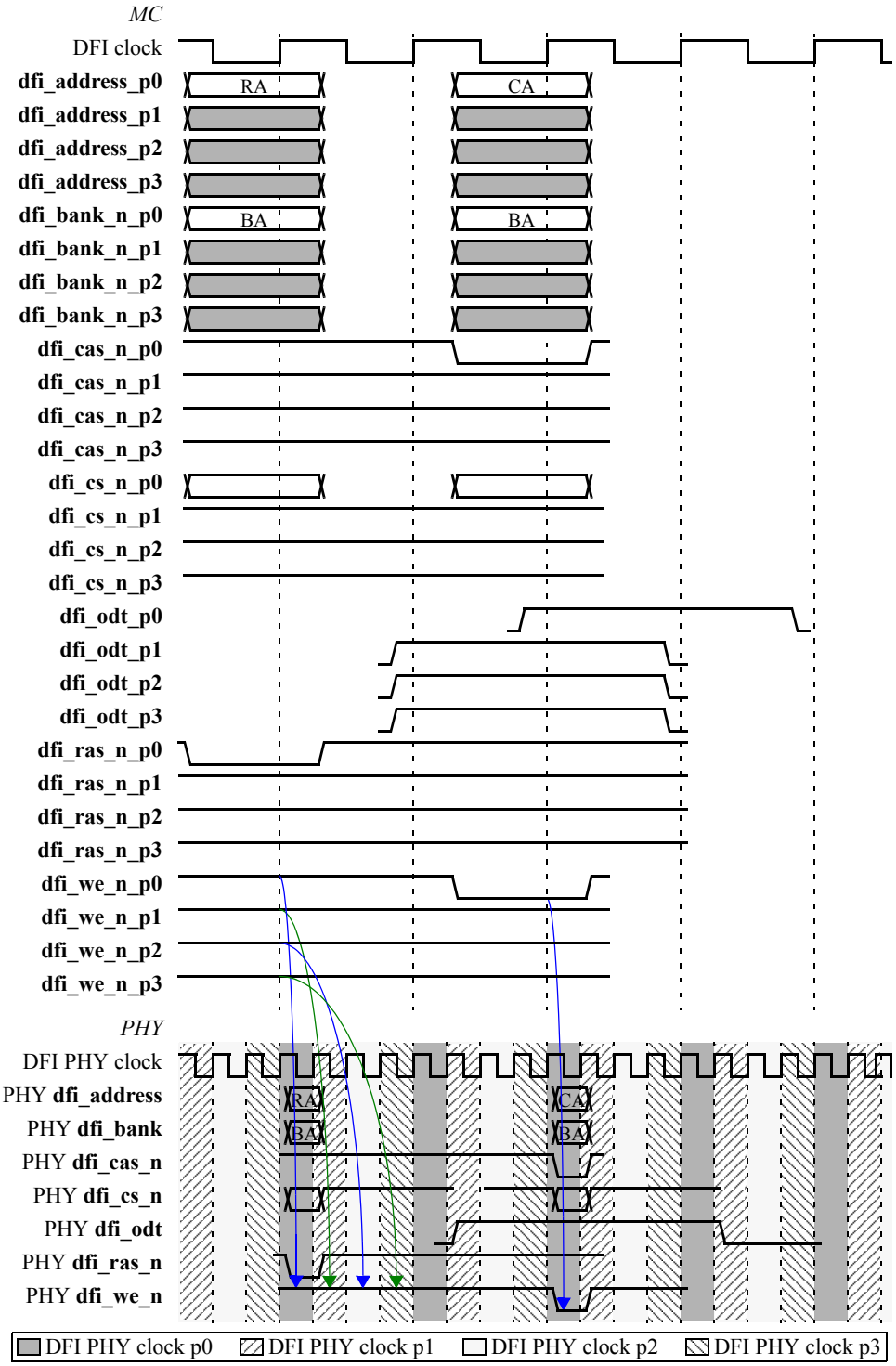


Figure 27 represents the same example, in a 1:4 frequency ratio system. The command is only sent on phase 0 and ODT information is provided on all phases. PHY timing is shown for illustrative purposes only.

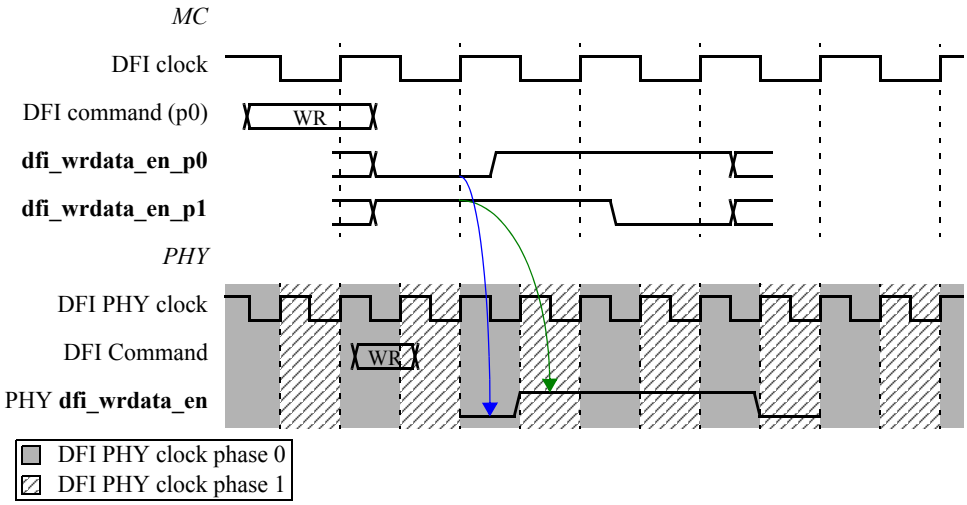
FIGURE 27. *Example 1:4 Frequency Ratio Command Stream*



4.7.2.1 Write Data Interface

The write data enable signal (**dfi_wrdata_en_pN**) defines the number of clocks between the write command and the write data, and its width defines the number of data phases of the write. In order to communicate this information to the PHY, the phase information must be encoded within the signal. Therefore, this signal will also be vectored into multiple signals based on the frequency ratio. Similar to the DFI command, each signal will be associated with a phase of the DFI PHY clock. Figure 28 demonstrates how a vectored **dfi_wrdata_en_pN** signal will be interpreted by the PHY in a 1:2 frequency ratio system. PHY timing is shown for illustrative purposes only.

FIGURE 28. *1:2 Frequency Ratio Write Data Example*



For matched frequency systems, the DFI write data bus width is generally twice the width of the DRAM data bus. For frequency ratio systems, this DFI write data bus width will be proportional to the frequency ratio to allow all of the write data that the memory requires to be sent in a single DFI clock cycle. The write data must be delivered with the DFI data words being aligned in ascending order.

The timing parameters $\mathbf{t_{phy\_wrlat}}$ and $\mathbf{t_{phy\_wrdata}}$ apply in frequency ratio systems in the same way as in matched frequency systems. These timing parameters define the delay from the write command to the **dfi_wrdata_en_pN** signal, and from the **dfi_wrdata_en_pN** signal to when data will be driven on the **dfi_wrdata_pN** signal, respectively. These timing parameters are defined in terms of DFI PHY clocks and are measured relative to how the PHY interprets the data.

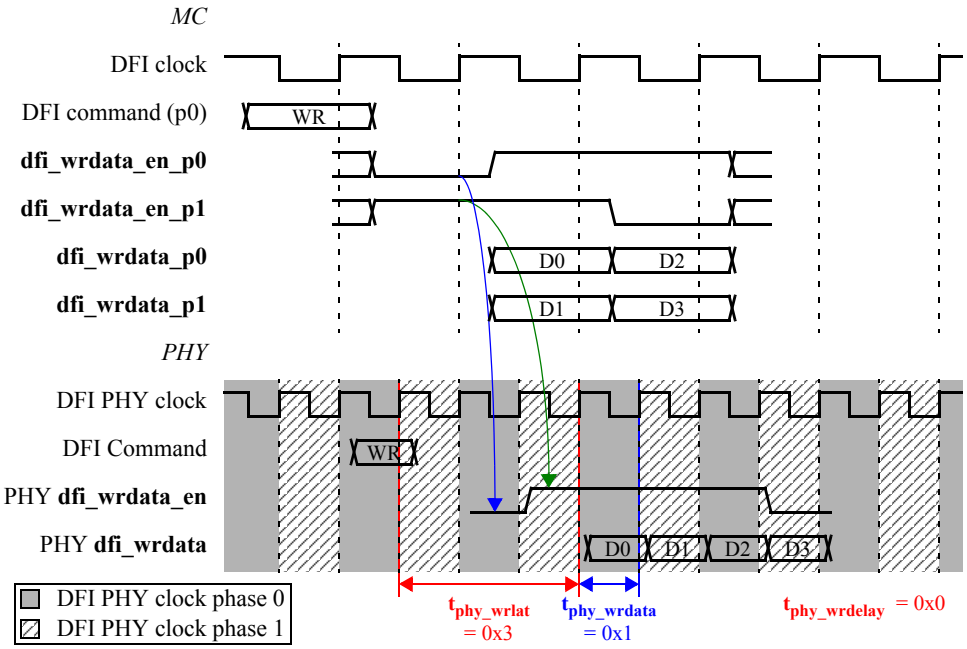
For frequency ratio systems, an additional timing parameter is included in the DFI specification to adjust the PHY write latency to match DRAM specifications. If the write data does not align to a rising edge of the DFI clock, either the MC will need to send the data un-aligned, or the PHY will need to add a cycle (or more) of delay. The

$t_{phy\_wrdelay}$ parameter is provided to adjust PHY timing. By setting this parameter to a non-zero value, the PHY will need to delay the write data captured from the DFI bus.

A PHY in a 1:2 frequency ratio system must support a $t_{phy\_wrdelay}$ parameter delay of 0 or 1. A PHY in a 1:4 frequency ratio system must support a $t_{phy\_wrdelay}$ parameter delay of 0, 1, 2 or 3. For all of the following figures, PHY timing is internal, and is only shown for illustrative purposes. Figure 29 and Figure 30 demonstrate systems where $t_{phy\_wrdelay}$ is set to 0. Figure 31 demonstrates the use of the $t_{phy\_wrdelay}$ timing parameter.

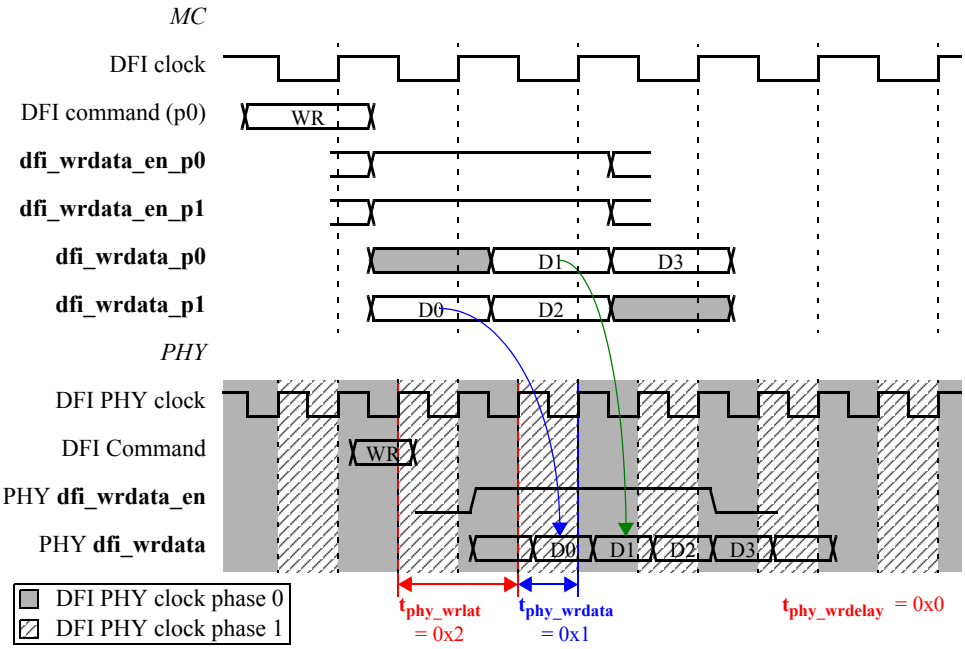
Figure 29 shows how data is received by the PHY in a situation where the data is sent aligned, but the enable signals are not aligned.

FIGURE 29. *1:2 Frequency Ratio Aligned Write Data Example*



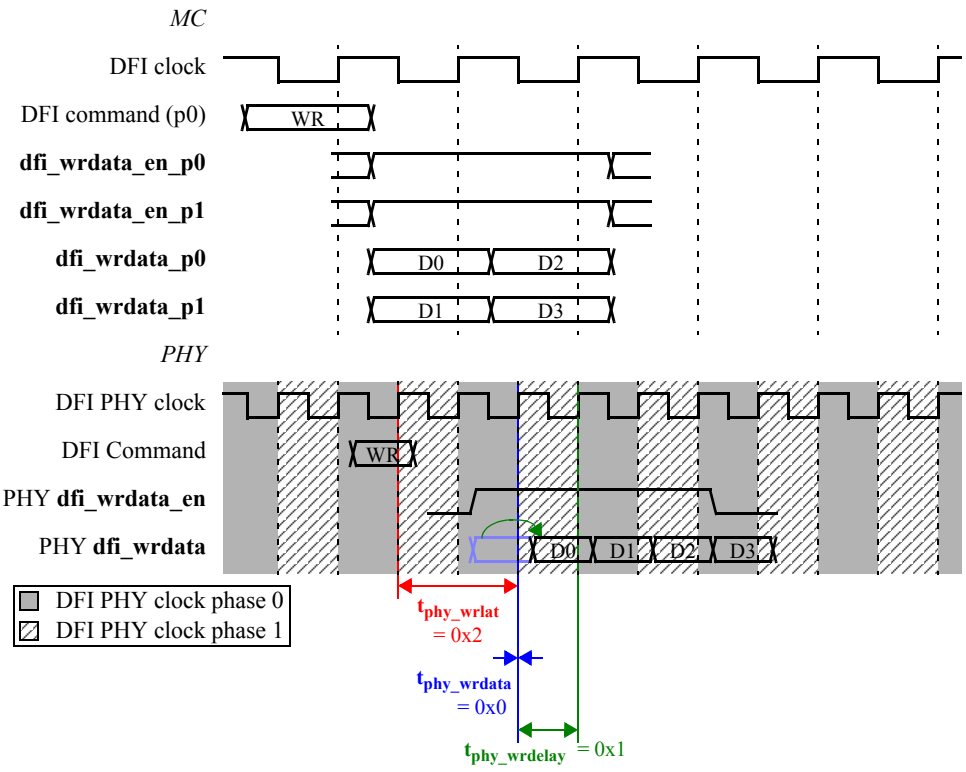
In Figure 30, the enable signals are sent aligned, but the data is not aligned. The MC sends the first beat of data on the phase 1 data signal.

FIGURE 30. *1:2 Frequency Ratio Aligned Write Enable Example*



As Figure 30 shows, the write data must be sent un-aligned to achieve the proper relationship between the command and data. Alternatively, the MC may send the data aligned (as shown in Figure 29), and set the $\mathbf{t_{phy\_wrdelay}}$ parameter to a non-zero value. This parameter adds additional delay in the PHY to the write data relative to the enable. Figure 31 shows how the $\mathbf{t_{phy\_wrdelay}}$ parameter can be used to adjust the timing inside the PHY.

FIGURE 31. *1:2 Frequency Ratio Aligned Write Enable and Write Data Example Using* $t_{phy\_wrdelay}$



4.7.2.2 Read Data Interface

Similar to the write enable signal, the read data enable signal (**dfi_rddata_en_pN**) defines the number of clocks between the read command and the read data, and its width defines the number of data phases of the read. The PHY will send read data to the MC on the **dfi_rddata_wN** buses whenever it is available, asserting the associated **dfi_rddata_valid_wN** signals to inform the MC which buses contain valid data. Unlike the read data enable signal which correlates to the phase of the DFI PHY clock, the read data, read data valid and read data not valid signals are all vectored with the DFI data word suffix.

For a 1:2 frequency ratio system, the read data bus is divided into 2 DFI read data words. For a 1:4 frequency ratio system, the read data bus is divided into 4 DFI read data words. Each DFI data word transfers a memory data word, the data associated with one rising and falling DQS. For example, in a 1:4 system with a memory data width of 32 bits, the DFI read data bus would consist of 4 64-bit DFI data words.

On a DFI clock, the PHY is permitted to assert any number of consecutive **dfi_rddata_valid_wN** signals that correspond to valid read data. However, the read data must be returned in a rolling order of DFI data words. For a 1:4 frequency ratio system, if read data is returned on the **dfi_rddata_w0** and **dfi_rddata_w1** buses on one

DFI clock cycle, the next transaction must return data starting on the **dfi_rddata_w2** bus, regardless of the number of DFI data words being returned. If that next transaction returned 2 DFI data words, data must be returned on the **dfi_rddata_w2** and **dfi_rddata_w3** buses. If that next transaction returned 4 DFI data words, data must be returned on the **dfi_rddata_w2**, **dfi_rddata_w3**, **dfi_rddata_w0** and **dfi_rddata_w1** buses - in that order.

For a 1:2 frequency ratio system, read data must be returned in the same manner, in a rolling order of DFI data words. In this case, there are only 2 DFI data words in the DFI read data bus - **dfi_rddata_w0** and **dfi_rddata_w1**.

This ordering rule must be followed regardless of whether the subsequent data transfer occurs on the next DFI clock or several clocks later. For LPDDR2 memory systems, this applies to both reads and mode register reads. The order is critical for the PHY and MC to correctly communicate read data. Each DFI data word must be used prior to sending data on the subsequent DFI data word, requiring data to be sent contiguously. The subsequent read data must be returned on the next DFI data word relative to the previous transaction. If the last transaction ended on **dfi_rddata_w2**, for example, the next transfer must begin on **dfi_rddata_w3**. Similarly, it is not legal to return read data on only the **dfi_rddata_w0** and **dfi_rddata_w2** buses.

Both the MC and the PHY must track which signals were used in the last transfer in order to interpret the data accurately. At initialization, the DFI data word pointer will be set to 0, and the first read data returned will be expected on the **dfi_rddata_w0** bus. During normal operation, certain procedures may affect the read data rotation, such as frequency changing, data training or gate training. Therefore, any assertion of the **dfi_init_start** signal, or a de-assertion of the **dfi_rdlvl_en** or **dfi_rdlvl_gate_en** signals must trigger a re-initialization of the DFI data word pointer to 0.

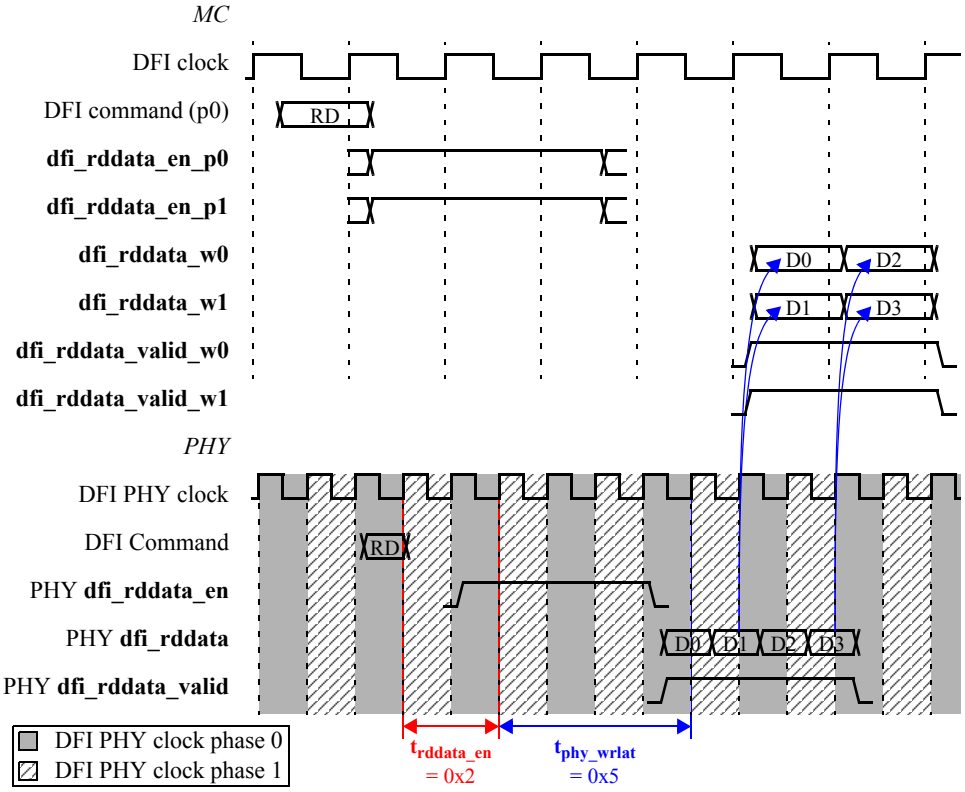
The rotational use of the **dfi_rddata_valid_wN** signals will only be required in situations where the system may return less data than the DFI read data bus. If the minimum transfer size is a multiple of the DFI read data bus width, data can always be returned on all DFI data words and the **dfi_rddata_valid_wN** signals will all be driven identically. Otherwise, only certain DFI data words of the DFI read data bus will be used. In either case, the MC must be able to receive data in a rotating order based on the last transfer to be DFI compliant for frequency ratio. A PHY may optionally be implemented such that it will always return read data on the entire DFI read data bus per transaction.

Regardless of how the signals are vectored, the PHY may only change read data, read data valid and read data not valid signals during phase 0 of the DFI PHY clock to allow the MC the entire DFI clock period to capture the signal and read data.

The timing parameters $\mathbf{t_{rddata\_en}}$ and $\mathbf{t_{phy\_rdlat}}$ apply in frequency ratio systems in the same way as in matched frequency systems. These timing parameters define the delay from the read command to the **dfi_rddata_en_pN** signal, and from the **dfi_rddata_en_pN** signal to when data will be returned on the **dfi_rddata_wN** bus, respectively. These timing parameters are defined in terms of DFI PHY clocks and are measured relative to how the PHY interprets the data.

Figure 32 demonstrates how a vectored **dfi_rddata_en_pN** signal will be interpreted by the PHY in a 1:2 frequency ratio system where all DFI data words are being returned on a DFI clock cycle. PHY timing is shown for illustrative purposes only.

FIGURE 32. *1:2 Frequency Ratio Single Read Data Example*



Similarly, Figure 33 demonstrates how a vectored **dfi_rddata_en_pN** signal will be interpreted by the PHY in a 1:4 frequency ratio system where all DFI data words are being returned on a DFI clock cycle. PHY timing is shown for illustrative purposes only.

FIGURE 33. *1:4 Frequency Ratio Single Read Data Example*

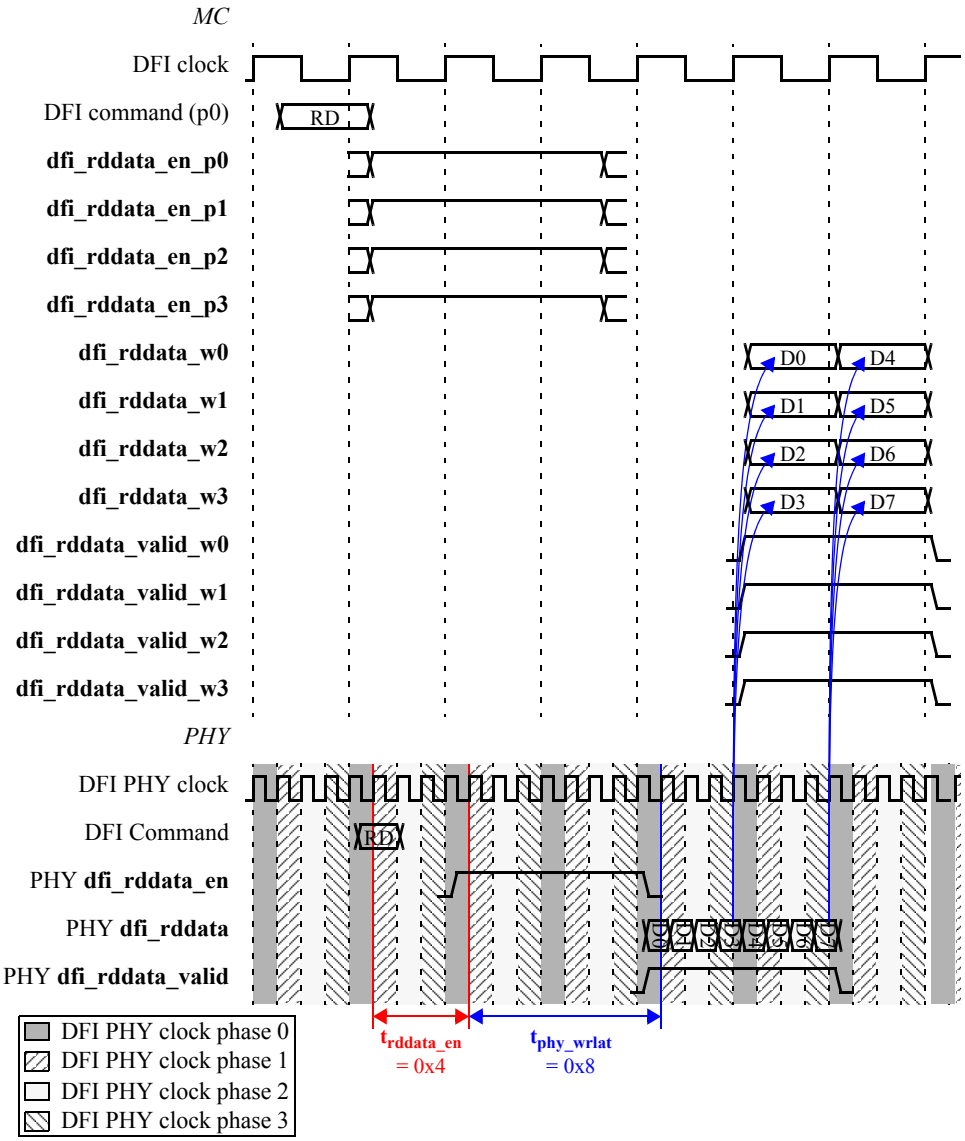
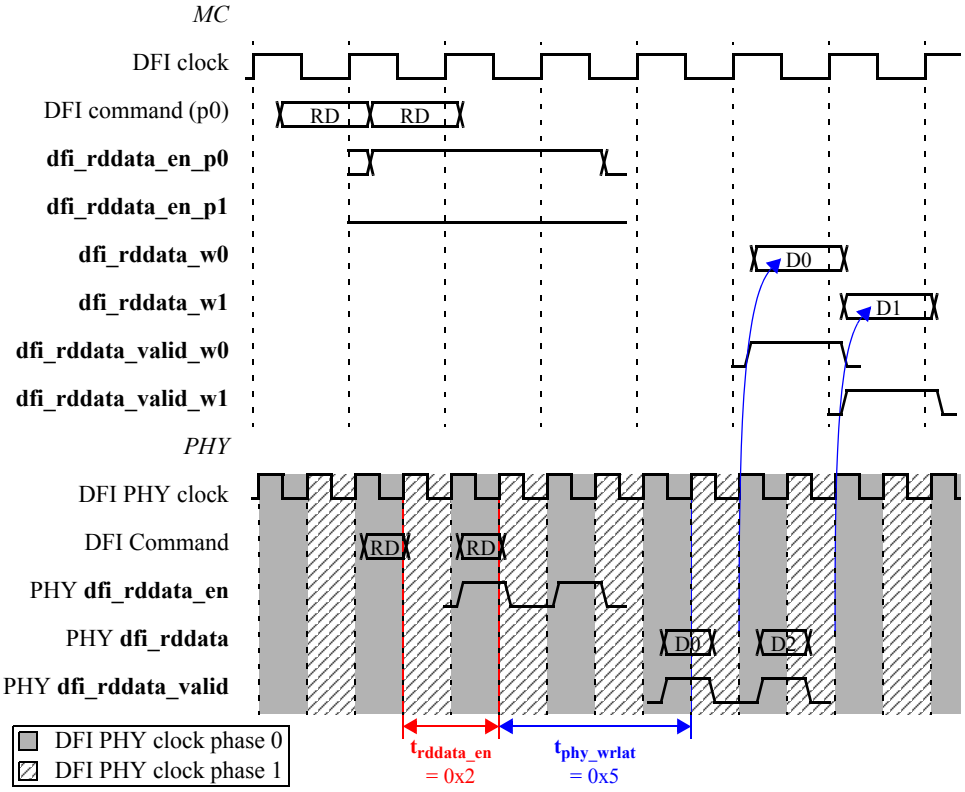


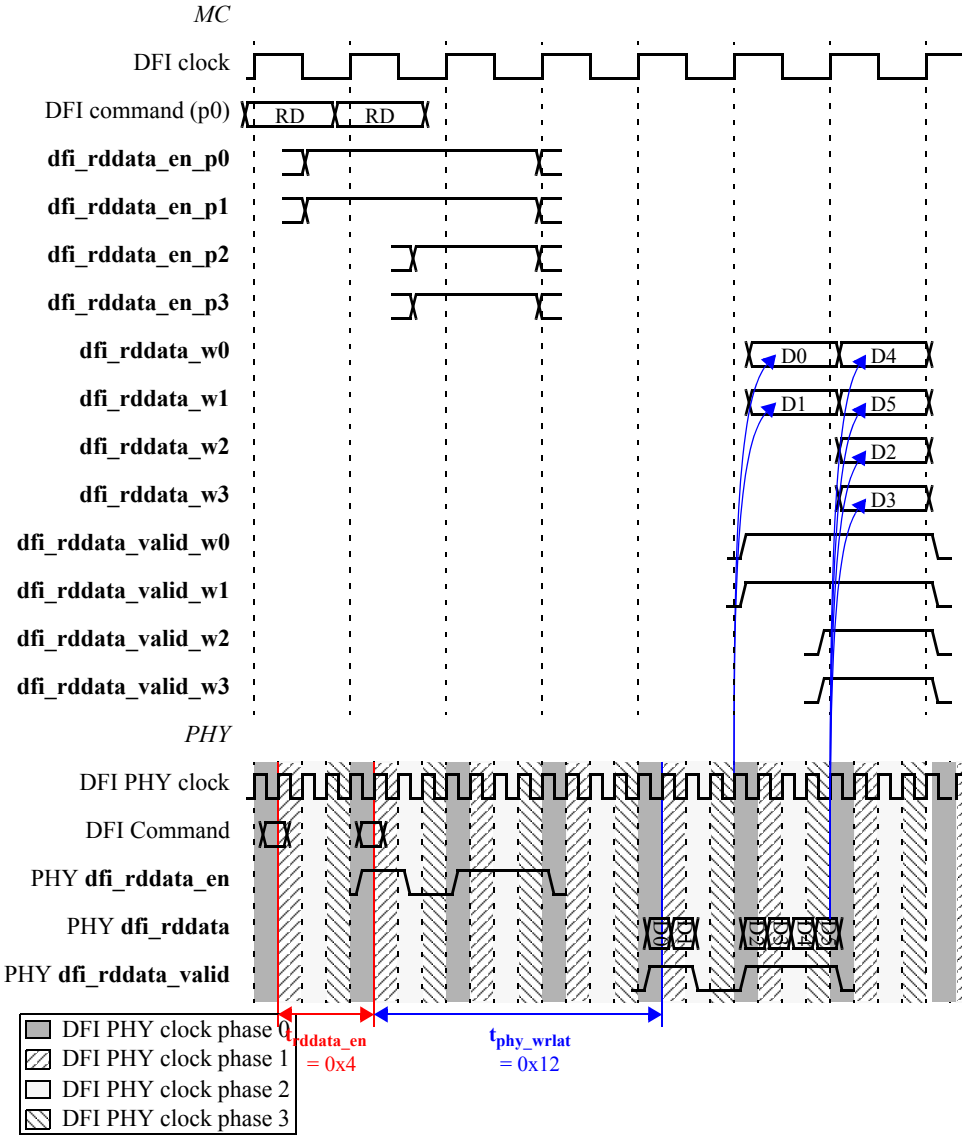
Figure 34 returns a single DFI data word with each command. The data for the second read command is returned on the **dfi_rddata_w1** bus following the rotational order rule. PHY timing is shown for illustrative purposes only.

FIGURE 34. *1:2 Frequency Ratio Multiple Read Data Example*



Similarly, Figure 35 shows a burst length 4 followed by a burst length 8 read. The data for the burst length 8 read command is returned starting on the **dfi_rddata_w3** bus following the rotational order rule. PHY timing is shown for illustrative purposes only.

FIGURE 35. *1:4 Frequency Ratio Multiple Read Data Example*



For matched frequency systems, the DFI read data bus width is generally twice the width of the DRAM data bus. For frequency ratio systems, this DFI read data bus width will be proportional to the frequency ratio to allow all of the read data that the memory returns to be sent in a single DFI clock cycle. The read data must be delivered with the DFI data words being aligned in ascending order.

4.7.2.3 Other Interfaces

The frequency ratio protocol does not affect the update, status or training interfaces. All of these signals will not be defined on a per-phase basis and all signal timing will be in reference to the DFI clock. The PHY must account for any assertions based on the DFI clock. Any signals driven by the PHY must only change during phase 0 of the DFI PHY clock to allow the MC the full DFI clock to capture the signal change.

4.8 Frequency Changing

There are situations in which the system may wish to change the clock frequency of the memory controller and PHY without completely resetting the system. The memory specifications define various memory states in which the clock frequency can be changed safely. The general procedure is to put the memory in one of these states, modify the clock frequency and then re-synchronize the system. When the new clock frequency has been established, the PHY may need to re-initialize various circuits to the new clock frequency prior to resuming normal memory operation. Once complete, the memory system is ready to resume normal operation. The DFI specification defines a frequency change protocol between the MC and the PHY to allow the devices to coordinate this frequency change process.

This is an optional feature of the DFI 2.1 specification and is not required for DFI compliance. The system may use a non-DFI frequency change method, or may choose to not support frequency change at all. However, if both the MC and the PHY intend to use the DFI 2.1 frequency change protocol, then they must comply with the handshaking defined by the specification. The handshaking protocol defines the signals through which the MC and the PHY will allow a frequency change to occur and also provides a means to abort the process if the PHY does not respond to a frequency change request. When a frequency change occurs, some of the DFI timing parameters may need to be changed.

NOTE: During the frequency change, the DFI clock must remain valid - either operating at a valid frequency or gated high or low.

The signals used in the frequency change protocol are **dfi_init_start** and **dfi_init_complete**. For more information on these signals, refer to Section 3.5, “Status Interface”.

4.8.1 Frequency Change Protocol - Acknowledged

During normal operation, once the **dfi_init_start** and **dfi_init_complete** signal have been asserted, the system may wish to change the DFI clock frequency. The MC asserts the **dfi_init_start** signal to indicate that a clock frequency change is being requested. The PHY should not interpret the initial **dfi_init_start** assertion as a frequency change request.

The MC guarantees that the **dfi_init_start** signal will remain asserted for at least $\mathbf{t_{init\_start}}$ cycles, allowing the PHY time to respond. The PHY may respond or ignore

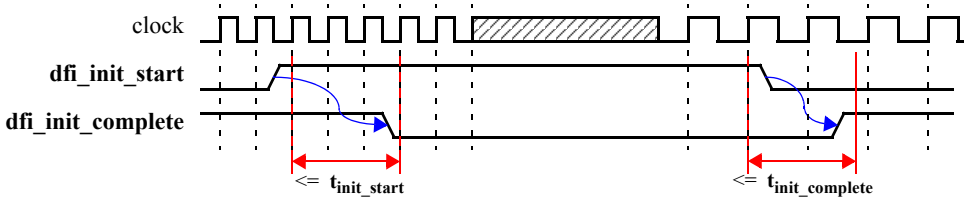
the frequency change request. To acknowledge the request, the **dfi_init_complete** signal must be de-asserted within $\mathbf{t_{init\_start}}$ cycles of the assertion of the **dfi_init_start** signal. The **dfi_init_complete** signal must de-assert at least one cycle before $\mathbf{t_{init\_start}}$ expires.

If the frequency change is acknowledged, the MC must hold the **dfi_init_start** signal asserted as long as the frequency change continues. Once the frequency change has completed, the MC will de-assert the **dfi_init_start** signal. The PHY must then complete any re-initialization required for the new clock frequency and re-assert **dfi_init_complete** within $\mathbf{t_{init\_complete}}$ cycles. This scenario is shown in Figure 36.

During a frequency change operation, the PHY must ensure that the memory interface is maintained at valid and stable levels throughout the operation to ensure that memory protocol is being observed. The MC must also insure that it maintains valid and stable levels on the DFI while **dfi_init_start** is asserted or **dfi_init_complete** is de-asserted.

Note that no maximum number of cycles for the entire cycle to complete is specified by the DFI.

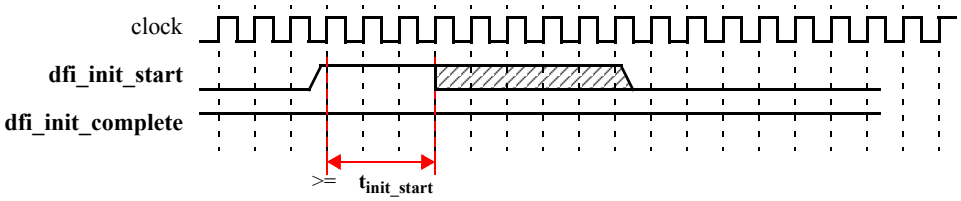
FIGURE 36. *Frequency Change Acknowledge Timing Diagram*



4.8.2 Frequency Change Protocol - Not Acknowledged

It is important to note that the PHY is not required to respond to a frequency change request. The MC must assert the **dfi_init_start** signal for at least $\mathbf{t_{init\_start}}$ cycles. This scenario is shown in Figure 37.

FIGURE 37. *Frequency Change Request Ignored Timing Diagram*



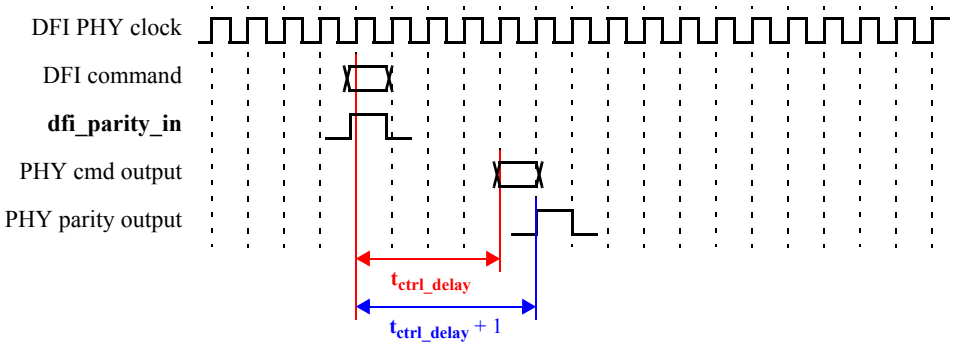
4.9 Parity Information

Parity bits are used in command transmission to verify that the command has been transmitted correctly between master and slave. A single parity bit is sent with each command and identifies if the number of bits set high in the **dfi_address**, **dfi_bank**, **dfi_cas_n**, **dfi_ras_n** and **dfi_we_n** signals is an even or an odd number.

If the DRAM receives a command where the number of bits of these signals set to ‘b1 does not match the even/odd setting of the **dfi_parity_in** signal, then an error occurred during transmission. This signal is sent from the MC simultaneously with the command and is valid for 1 cycle. The command and other DFI signals are reflected on the PHY/DRAM interface $\mathbf{t_{ctrl\_delay}}$ cycles after being sent.

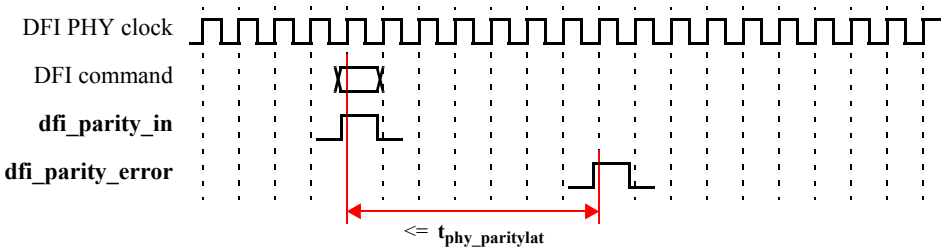
However, to meet DIMM timing requirements, the parity signal from the PHY to the DRAM must be sent one cycle after the command output. Therefore, the PHY is required to add one cycle of latency to the **dfi_parity_in** signal before it is reflected to the DIMMs. This requirement is shown in Figure 38 with an odd command parity example.

FIGURE 38. *Odd Command Parity Example Timing Diagram*



DDR2 and DDR3 registered DIMMs compute parity on the incoming command and compare the computed value with the value driven in on the DFI parity signal. If these values do no match, the DIMM asserts a parity error output, which is sent back to the PHY on the PAR_IN signal and then to the MC on the **dfi_parity_error** signal. The **dfi_parity_error** signal will be asserted within $\mathbf{t_{phy\_paritylat}}$ cycles of the associated **dfi_parity_in** signal. Since $\mathbf{t_{phy\_paritylat}}$ is a maximum value, the **dfi_parity_error** signal can not be correlated to one specific command, but to any command sent within the last $\mathbf{t_{phy\_paritylat}}$ cycles. The timing between the command and error is shown in Figure 39 with an odd command parity example.

FIGURE 39. *Odd Command Parity Error Example Timing Diagram*



4.10 Training Operations - Read and Write Leveling

The DFI contains a set of signals to support read and write leveling. The read leveling logic may be used to perform both data eye training and gate training.

4.10.1 Read Leveling

The goal of data eye training is to identify the delay at which the read DQS rising edge aligns with the beginning and end transitions of the associated DQ data eye. By identifying these delays, the system can calculate the midpoint between the delays and accurately center the read DQS within the DQ data eye. Data eye training is used by DDR3 and LPDDR2 memory systems.

The goal of gate training is to locate the delay at which the initial read DQS rising edge aligns with the rising edge of the read DQS gate. Once this point is identified, the read DQS gate can be adjusted prior to the DQS, to the approximate midpoint of the read DQS preamble. The gate training operation requires that the read DQS gate be placed within the bounds of the beginning of the read DQS preamble and the falling edge of the first read DQS for the response to properly indicate the alignment of gate to the first read DQS. Another method may be necessary to locate the read DQS gate within this timing window. Gate training is expected to be run iteratively to validate that the gate has been properly placed.

For data eye training, the signals used are: **dfi_rdlvl_en**, **dfi_rdlvl_req**, **dfi_rdlvl_load**, **dfi_rdlvl_resp**, **dfi_rdlvl_cs_n**, **dfi_rdlvl_delay_X**, **dfi_rdlvl_mode** and **dfi_rdlvl_edge**. LPDDR2 memory systems also use the **dfi_rddata_valid** and **dfi_rddata** signals. For more information on these signals, refer to Section 3.3, “Read Data Interface” and Section 3.6, “Training Interface”.

For gate training, the signals used are: **dfi_rdlvl_gate_en**, **dfi_rdlvl_gate_req**, **dfi_rdlvl_load**, **dfi_rdlvl_resp**, **dfi_rdlvl_cs_n**, **dfi_rdlvl_gate_delay_X**, **dfi_rdlvl_gate_mode** and **dfi_rdlvl_edge**. For more information on these signals, refer to Section 3.6, “Training Interface”.

Figure 40 and Figure 41 demonstrate how the response signal is related to the gate. When the gate rises when DQS is low, a “0” response is sent. When the gate rises when

DQS is high, a “1” response is sent. By adjusting the delay, the system will be able to capture when the transition occurs which identifies when the gate is aligned to the first rising edge of the DQS.

FIGURE 40. *Gate Leading DQS Timing Diagram*

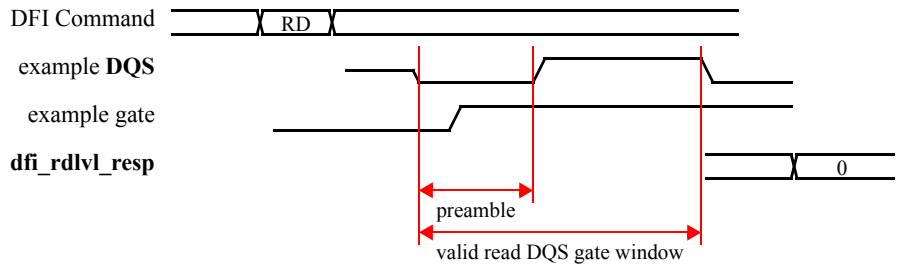
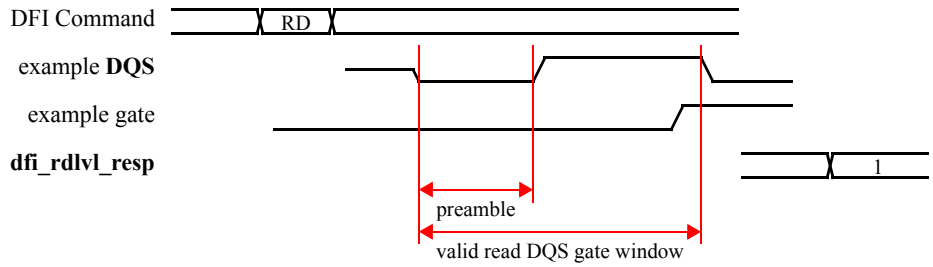


FIGURE 41. *Gate Lagging DQS Timing Diagram*



4.10.2 Write Leveling

The goal of write leveling is to locate the delay at which the write DQS rising edge aligns with the rising edge of the memory clock. By identifying this delay, the system can accurately align the write DQS within the memory clock. Write leveling is specific to DDR3 memory systems.

The signals used in write leveling are: **dfi_wrlvl_en**, **dfi_wrlvl_req**, **dfi_wrlvl_load**, **dfi_wrlvl_strobe**, **dfi_wrlvl_resp**, **dfi_wrlvl_cs_n**, **dfi_wrlvl_delay_X** and **dfi_wrlvl_mode**. For more information on these signals, refer to Section 3.6, “Training Interface”.

4.10.3 Training Interface Operating Modes

The DFI defines four operating modes for the training interface for data eye training, gate training and write leveling. Data eye training and gate training apply to both DDR3 and LPDDR2 memory systems. DDR3 memory systems also support write leveling. For DFI compliance, MC must support all four operating modes of the given training operation, while the PHY must support at least one operating mode. Most PHYs will

support the same mode for all training operations, but this is not a requirement. The modes are:

- No Support
- MC Evaluation
- PHY Evaluation
- PHY Independent

These modes define whether the MC or PHY, or neither, maintains the responsibility for managing the programming of the delay lines and evaluating of the response. In MC Evaluation or PHY Evaluation modes, the MC will generate the MRS commands, assert the enable signal, and generate the read commands (DDR3), mode register read commands (LPDDR2) or write strobes (DDR3). During either read leveling operation in PHY Evaluation mode, the **dfi_rddata_valid** signal is ignored.

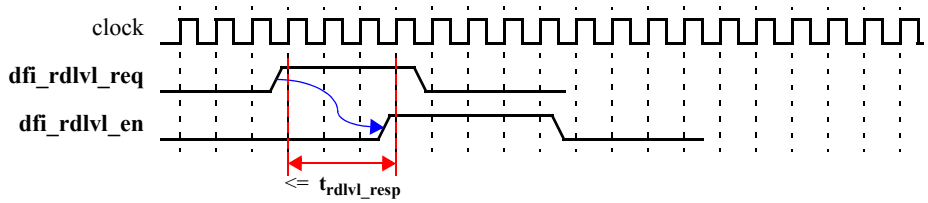
If any of the training operations are run in PHY Independent mode, the PHY will perform the associated training (data eye training, gate training or write leveling) without any support from the MC to perform the operation. For the training operation run in PHY Independent mode, only the mode indicator signal is used and all other signals of the training interface are irrelevant.

4.10.3.1 Initiating a Training Operation

For MC Evaluation or PHY Evaluation modes, any training may be initiated by software, by the MC or by the PHY. Training may be executed during initialization, frequency change or for tuning during normal operation. However, the PHY should not request any training during initialization. The MC is responsible for initiating any data eye training, gate training and/or write leveling required during initialization or frequency change unless the PHY is operating in PHY Independent mode.

The PHY can request data eye training by driving the **dfi_rdlvl_req** signal, write leveling by driving the **dfi_wrlvl_req** signal, or gate training by driving the **dfi_rdlvl_gate_req** signal. If the PHY is operating in PHY Evaluation mode for data eye training and gate training, these operations may be issued through either set of signals since the operations are identical from the MC perspective. The MC must respond to any of these requests by asserting the appropriate enable (**dfi_rdlvl_en**, **dfi_wrlvl_en** or **dfi_rdlvl_gate_en**) within the relevant $\mathbf{t_{rdlvl\_resp}}$ or $\mathbf{t_{wrlvl\_resp}}$ cycles. Figure 42 shows this timing relationship for the data eye training process. The timing is similar for gate training and write leveling.

FIGURE 42. *Data Eye Training Request Timing*



If the PHY uses the PHY Evaluation mode for training, the MC will wait for the response signal to be asserted before disabling the active logic. The response will always be received on the **dfi_rdlvl_resp** or **dfi_wrlvl_resp** signals. For this one case, the response will be seen on the **dfi_rddata** signal. The DFI specifies maximum times that the system will wait for a response on the **dfi_rdlvl_resp** or **dfi_wrlvl_resp** signals as $\mathbf{t_{rdlvl\_max}}$ and $\mathbf{t_{wrlvl\_max}}$, respectively.

4.10.3.2 MC Evaluation Mode

In MC Evaluation mode, the MC is responsible for adjusting the delay and evaluating the response sampled at the interface until the required delay values are determined. For data eye training, the response signal will be DQ sampled by DQS. For gate training, the response will indicate the location of the read DQS gate relative to DQS. For write leveling, the response will be identical to the response driven by the DRAM on the DQ bus during the write leveling command.

The logic will be used to locate the necessary edges and the MC will use this information to calculate and drive the delays. The MC will also control the enabling and disabling of the logic in the DRAMs and the PHY and generate the necessary read commands, mode register read commands or write strobes. The PHY logic is enabled/ disabled by the assertion/de-assertion of the **dfi_rdlvl_en**, **dfi_rdlvl_gate_en** and **dfi_wrlvl_en** signals.

The MC must complete all transactions in progress to memory prior to initiating any of the leveling operations. Once any of the enable signals are asserted, the PHY should immediately enable the associated logic.

For DDR3 memory systems, the MC uses the information passed back from the PHY on the **dfi_rdlvl_resp** signal or the **dfi_wrlvl_resp** signal to control leveling. The timing parameters $\mathbf{t_{rdlvl\_resplat}}$ or $\mathbf{t_{wrlvl\_resplat}}$ are used to determine the validity of the response data. The response must remain stable from when it becomes valid until either the next read command or write strobe is sent or the logic is disabled. For data eye training or gate training, the MC will send read commands, waiting at least $\mathbf{t_{rdlvl\_rr}}$ cycles between commands and at least $\mathbf{t_{rdlvl\_dll}}$ cycles after a load before issuing the next command. The results will be monitored to make adjustments for DQS and gate alignment. For write leveling, the MC will send write strobes, waiting at least $\mathbf{t_{wrlvl\_ww}}$ cycles between write strobes and at least $\mathbf{t_{wrlvl\_dll}}$ cycles after a load before issuing the next write strobe. The results will be monitored to make adjustments for write DQS

alignment. The **dfi_rdlvl_load** signal or **dfi_wrlvl_load** signal will be asserted for one clock to indicate when delays are updated.

For LPDDR2 memory systems, the MC uses the information passed back from the PHY on the **dfi_rddata** signal associated with the **dfi_rddata_valid** signal to control leveling. The MC will send mode register read commands, waiting at least $\mathbf{t_{rdlvl\_rr}}$ cycles between commands and at least $\mathbf{t_{rdlvl\_dll}}$ cycles after a load before issuing the next command. The results will be monitored to make adjustments for DQS and gate alignment. The **dfi_rdlvl_load** signal will be asserted for one clock to indicate when delays are updated.

When the MC has found the necessary edges and completed training, it de-asserts the enable signal (**dfi_rdlvl_en**, **dfi_rdlvl_gate_en** or **dfi_wrlvl_en**) and for DDR3 memory systems, sends an MRS command to disable the leveling logic in the DRAM. This completes training with the new delay values sent on the **dfi_rdlvl_delay_X**, **dfi_rdlvl_gate_delay_X** or **dfi_wrlvl_delay_X** signals.

4.10.3.2.1 Data Eye Training in MC Evaluation Mode

Figure 43 demonstrates data eye training in this mode for DDR3 memory systems. The MRS commands are used to enable and disable the data eye training logic in the DRAMs and the **dfi_rdlvl_en** signal is used to enable/disable the data eye training logic in the PHY.

Figure 44 demonstrates data eye training in this mode for LPDDR2 memory systems. LPDDR2 memories do not require an MRS command for setup. This memory requires mode register reads to the mode registers MR32 and MR40 to perform data eye training.

Once the logic is enabled, read commands (DDR3) or mode register read commands (LPDDR2) are issued regularly, obeying the timing parameters as shown. The responses are evaluated and the delays are adjusted based on the evaluation. This process may take several iterations of read commands. When the beginning and end transitions have been located and the midpoint has been calculated, the MC releases the **dfi_rdlvl_en** signal. This completes data eye training with the new delay values sent on the **dfi_rdlvl_delay_X** signals.

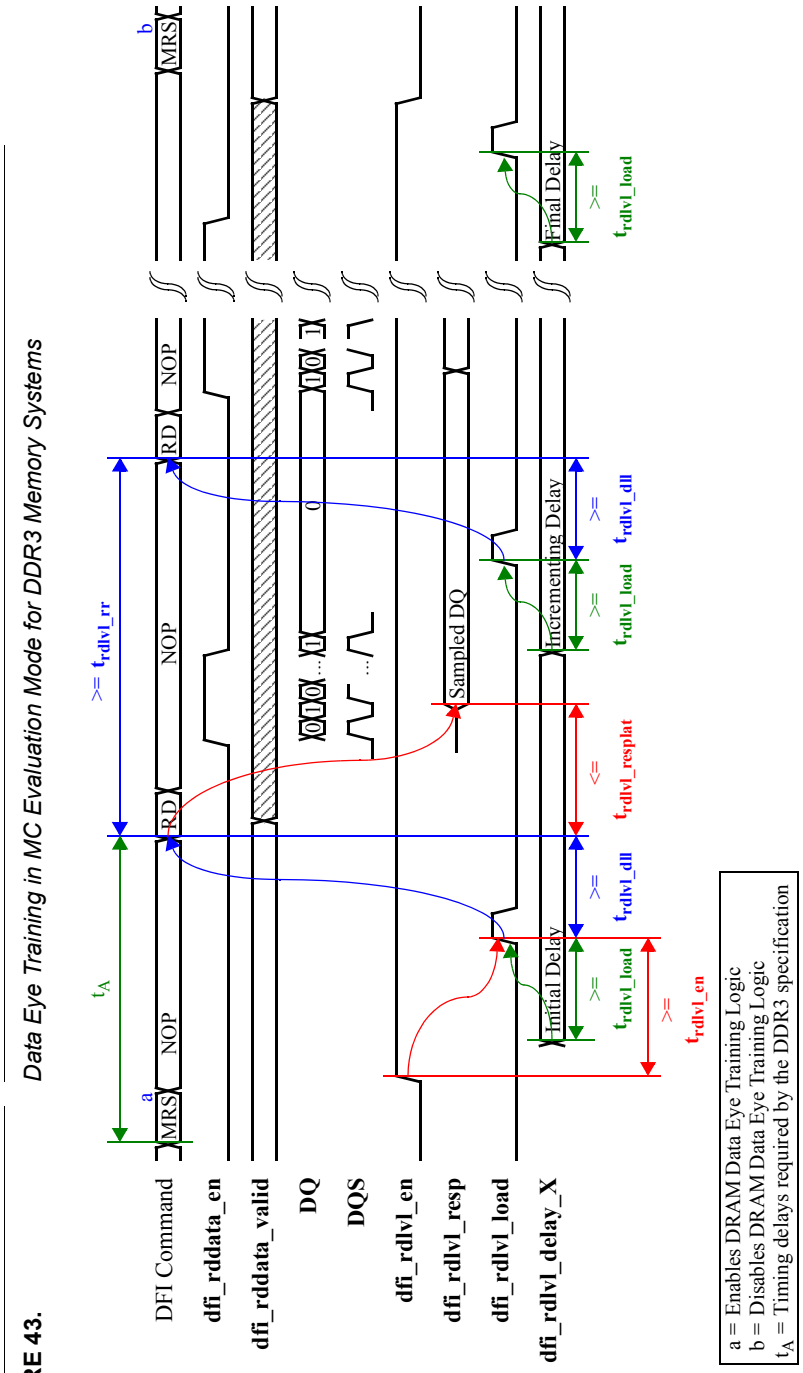


FIGURE 43. *Data Eye Training in MC Evaluation Mode for DDR3 Memory Systems*

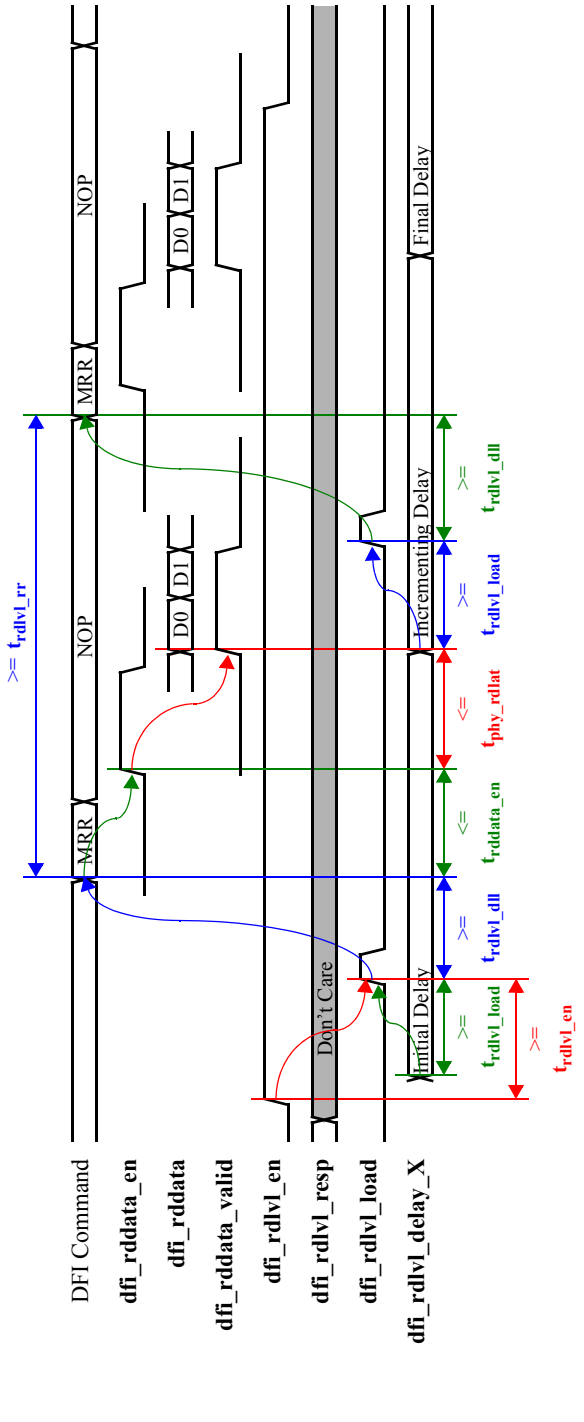


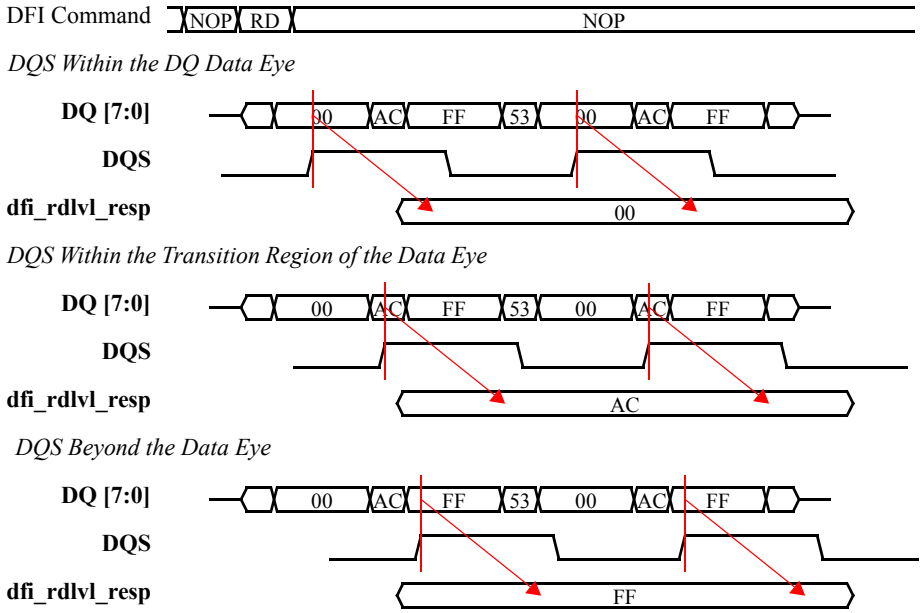
FIGURE 44. *Data Eye Training in MC Evaluation Mode for LPDDR2 Memory Systems*

In the MC Evaluation mode, the MC is required to analyze the data response from the transactions being executed. For this to be possible, the response information (the value of the **dfi_rdlvl_resp** signal or the **dfi_rddata** signal) must be clearly defined.

For DDR3 memory systems performing data eye training, responses are returned on the **dfi_rdlvl_resp** signal. The expected value of the **dfi_rdlvl_resp** signal is the DQ sampled by the rising edge of DQS as shown in Figure 45. The DQ may be sampled by $\overline{\text{DQS}}$, however, in this case, the PHY must invert the response prior to sending the value on the DFI signal to match the response shown in Figure 45. Since the memory may support sending a single DQ per memory or the entire DQ bus, the PHY must define the width of the response per data slice.

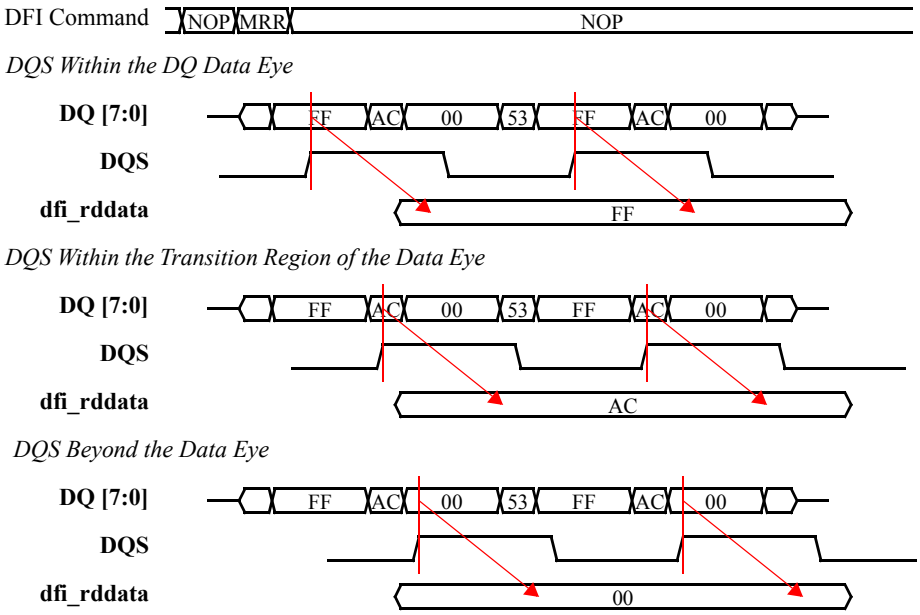
Figure 45 shows three scenarios of the relative relationship between the DQ and DQS. This example assumes an 8-bit DRAM with all DQ's returning a predefined pattern of "0-1-0-1." The **dfi_rdlvl_resp** signal reflects the value of the DQ signal at the DQS rising edge. The MC should use this information to locate the transition points. The figure shows data patterns of 0xAC and 0x53 as transitional values. These regions represent where the data response is uncertain due to skew, jitter or setup/hold timing violations.

FIGURE 45. *Data Eye Training Response for DDR3 Memory Systems for an 8-Bit Data Slice*



For LPDDR2 memory systems performing data eye training in MC Evaluation mode, responses are returned on the **dfi_rddata** signal. The expected value of the **dfi_rddata** signal is a burst length four response of four data phases defined by mode register reads of MR32 and MR40. Figure 46 shows three scenarios of the relative relationship between the DQ and DQS. This example assumes an 8-bit DRAM with all DQ's returning the predefined patterns of "1-0-1-0" or "0-0-1-1." The **dfi_rddata** signal reflects the value of the DQ signal at the DQS rising edge. The MC should use this information to locate the transition points. The figure shows data patterns of 0xAC and 0x53 as transitional values. These regions represent where the data response is uncertain due to skew, jitter or setup/hold timing violations.

FIGURE 46. *Data Eye Training Response for LPDDR2 Memory Systems in MC Evaluation Mode for an 8-Bit Data Slice*



4.10.3.2.2 Gate Training in MC Evaluation Mode

Figure 47 and Figure 48 demonstrates gate training in this mode for DDR3 and LPDDR2 memory systems, respectively. These timing diagrams are very similar to the timing diagrams for data eye training, with the exception that the **dfi_rdlvl_gate_en** signal is used to enable/disable the gate training logic in the PHY.

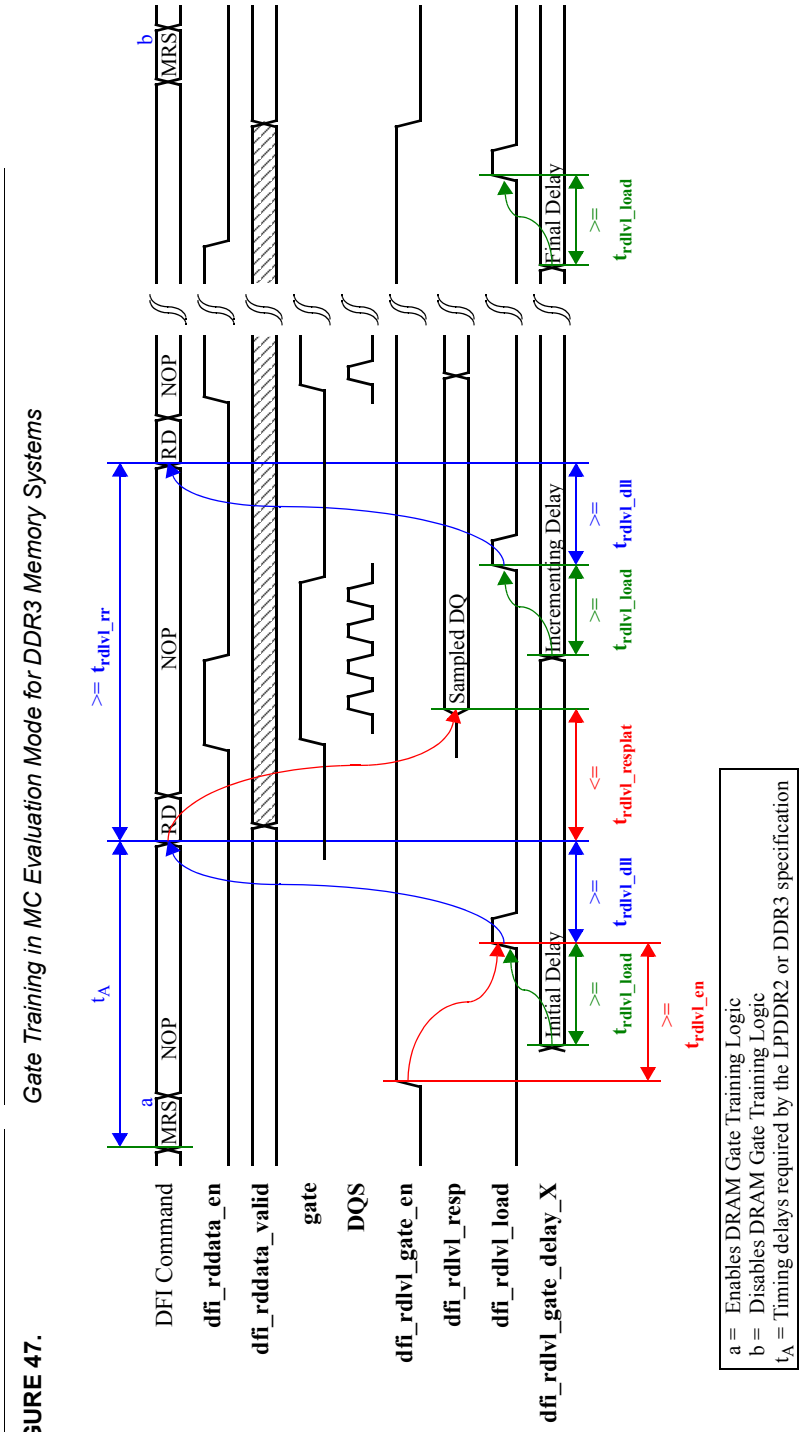


FIGURE 47. *Gate Training in MC Evaluation Mode for DDR3 Memory Systems*

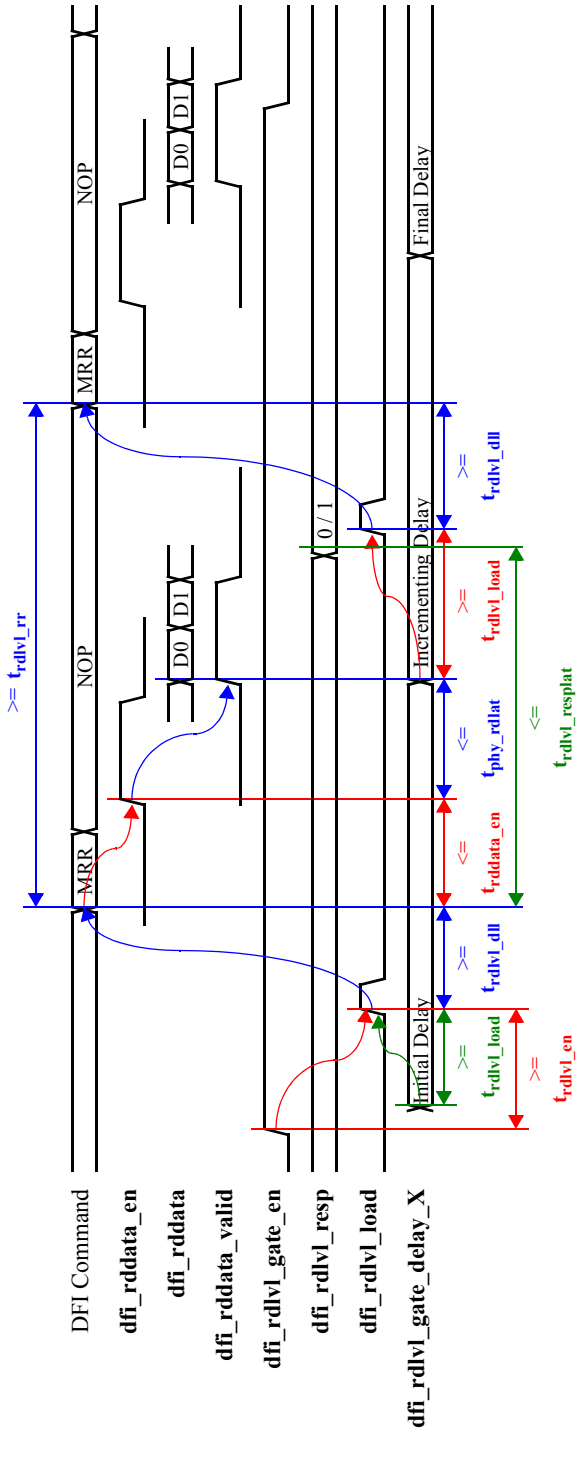
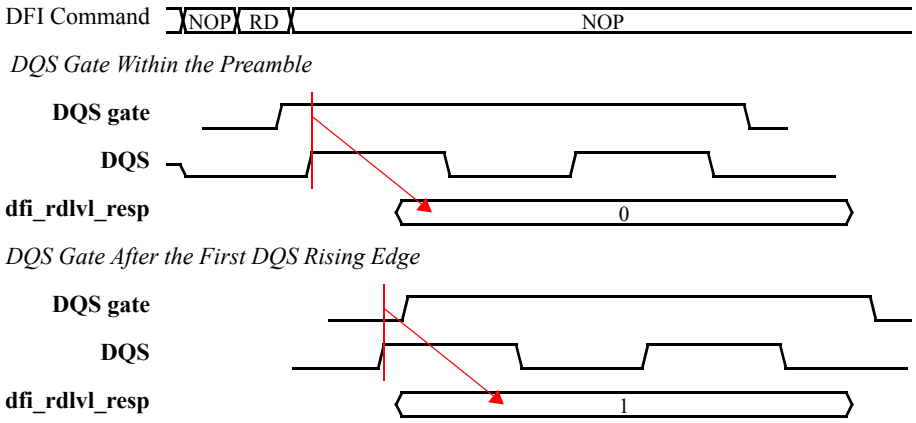


FIGURE 48. *Gate Training in MC Evaluation Mode for LPDDR2 Memory Systems*

Responses are returned on the **dfi_rdlvl_resp** signal. The **dfi_rdlvl_resp** signal must be the value of the read DQS at the rising edge of the read DQS gate of the read transaction as shown in Figure 49. The PHY must define the width of the response; a recommended width is 1 bit per data slice. For LPDDR2 memory systems, during gate training, the MRR data is returned to the MC on the **dfi_rddata** signal, but these values are discarded by the MC.

Figure 49 shows two scenarios of the relative relationship between the gate and DQS. The **dfi_rdlvl_resp** signal reflects the value of the DQS at the rising edge of the DQS gate. The MC should use this information to locate the transition point.

FIGURE 49. *Gate Training Response During Gate Training for a Single Data Slice*



4.10.3.2.3 Write Leveling in MC Evaluation Mode

Figure 50 demonstrates write leveling in this mode. The MRS commands are used to enable and disable the write leveling logic in the DRAMs and the **dfi_wrlvl_en** signal is used to enable/disable the write leveling logic in the PHY. Write leveling is specific to DDR3 memories.

Once the logic is enabled, write strobes are issued regularly, obeying the timing parameters as shown. Responses are returned on the **dfi_wrlvl_resp** signal. The delays are adjusted based on the evaluation. This process may take several iterations of write strobes. When the transition has been located, the MC releases the **dfi_wrlvl_en** signal. This completes write leveling with the new delay values sent on the **dfi_wrlvl_delay_X** signals.

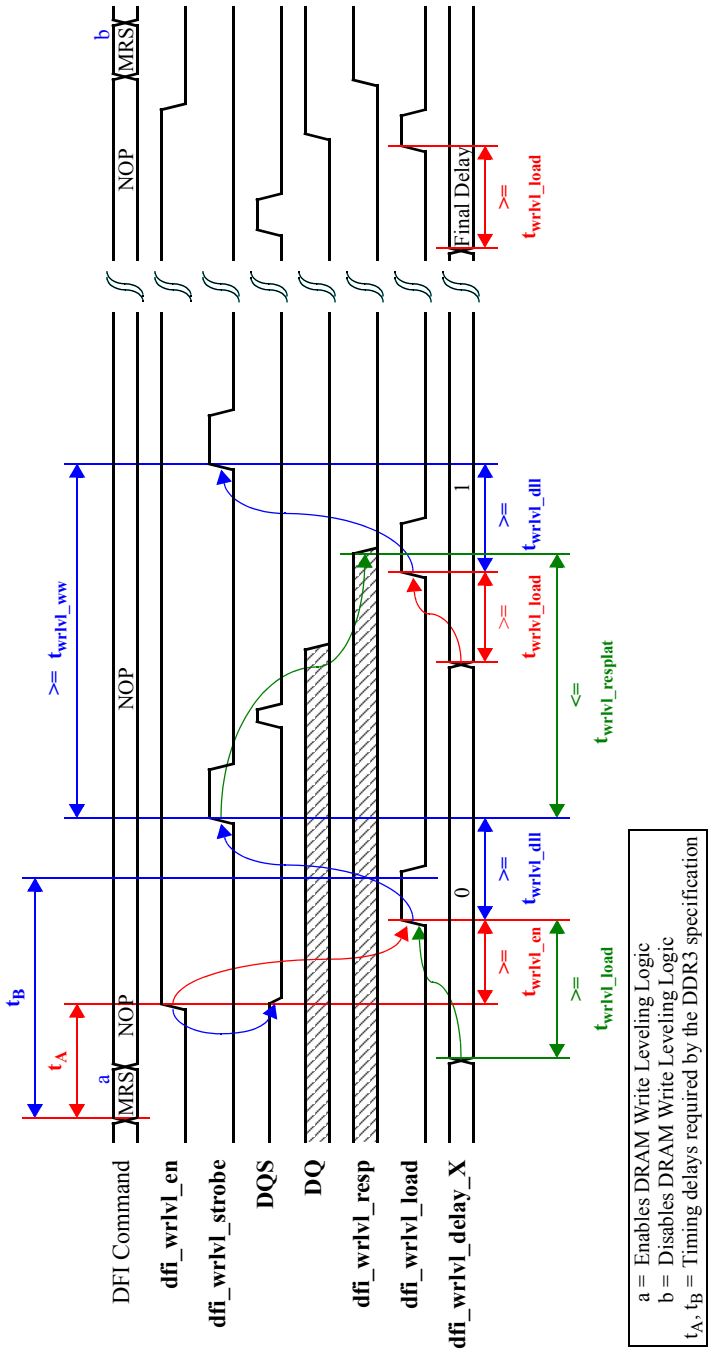


FIGURE 50. *Write Leveling in MC Evaluation Mode for DDR3 Memory Systems*

4.10.3.3 PHY Evaluation Mode

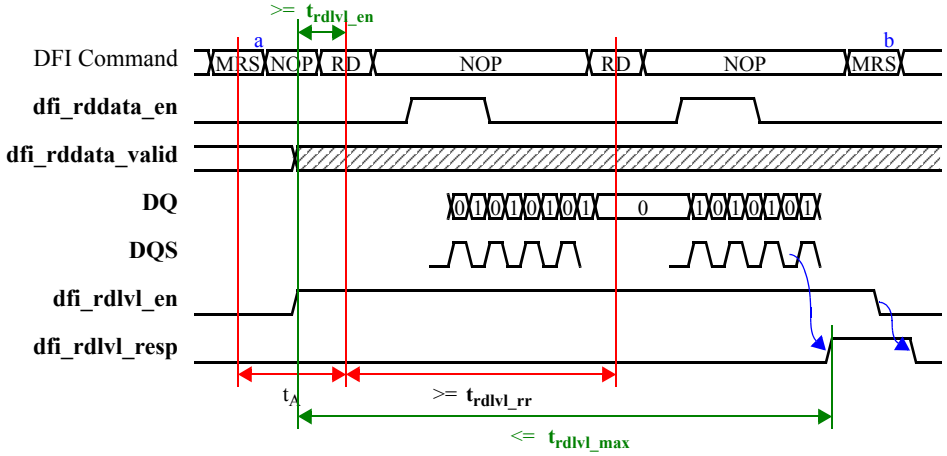
In PHY Evaluation mode, the PHY is responsible for determining the correct delay programming for the read data DQS, read DQS gate and write DQS signals. The PHY adjusts the delays and evaluates the results to locate the appropriate edges. The MC assists by enabling and disabling the leveling logic in the DRAMs and the PHY and by generating the necessary read commands, mode register reads or write strobes. The PHY informs the MC when it has completed training, which triggers the MC to stop generating commands and to return to normal operation.

The MC must complete all transactions in progress to memory prior to initiating any of the leveling operations. Once any of the enable signals are asserted, the PHY should immediately enable the associated logic. In PHY Evaluation mode, the MC will not receive the memory response from the PHY. Therefore the only relevant DFI timing parameters are $\mathbf{t_{rdlvl\_rr}}$ for read leveling and $\mathbf{t_{wrlvl\_ww}}$ for write leveling. The timing parameter $\mathbf{t_{rdlvl\_rr}}$ defines the minimum number of cycles that the MC should wait between issuing reads for DDR3 memory systems or mode register reads for LPDDR2 memory systems. The timing parameter $\mathbf{t_{wrlvl\_ww}}$ dictates the minimum delay between write strobes. The MC will continue to drive subsequent commands every $\mathbf{t_{rdlvl\_rr}}$ cycles, or subsequent write strobes every $\mathbf{t_{wrlvl\_ww}}$ until the PHY drives all bits of the response signal (**dfi_rdlvl_resp** or **dfi_wrlvl_resp**) high.

4.10.3.3.1 Read Leveling in PHY Evaluation Mode

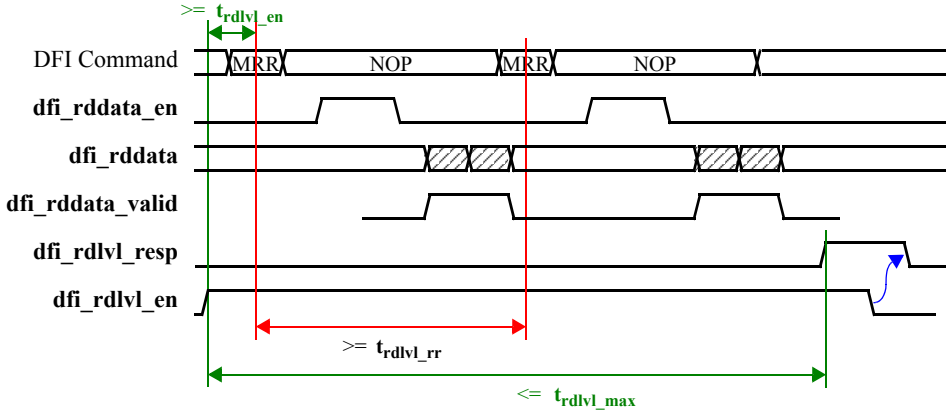
Figure 51 and Figure 52 demonstrates read leveling in this mode for DDR3 and LPDDR2 memory systems, respectively. For DDR3 memory systems, MRS commands are used to enable and disable the read leveling logic in the DRAMs. For LPDDR2 memory systems, no MRS commands are required to enable/disable the leveling mode in the DRAMs. In both cases, the **dfi_rdlvl_en** signal is used to enable/disable the read leveling logic in the PHY. All evaluations and delay changes are handled within the PHY. While the response data is returned on the **dfi_rddata** signal, it is not used by the MC. When the PHY has found the necessary edges and completed read leveling, it drives the **dfi_rdlvl_resp** signal high, which informs the MC that the procedure is done. The MC then de-asserts the **dfi_rdlvl_en** signal. Once the **dfi_rdlvl_en** signal has de-asserted, the PHY may stop driving the **dfi_rdlvl_resp** signal.

FIGURE 51. *Read Leveling in PHY Evaluation Mode for DDR3 Memory Systems*



a = Enables DRAM Read Leveling Logic
b = Disables DRAM Read Leveling Logic
t_A = Timing delays required by the DDR3 specification

FIGURE 52. *Read Leveling in PHY Evaluation Mode for LPDDR2 Memory Systems*



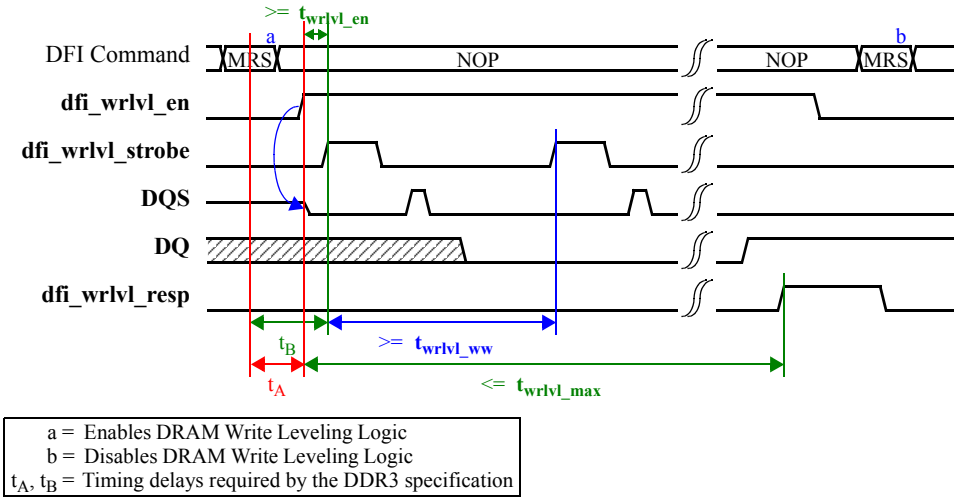
4.10.3.3.2 Write Leveling in PHY Evaluation Mode

Figure 53 demonstrates write leveling in this mode. The MRS commands are used to enable and disable the write leveling logic in the DRAMs and the **dfi_wrlvl_en** signal is used to enable/disable the write leveling logic in the PHY. Write leveling is specific to DDR3 memory systems.

All evaluations and delay changes are handled within the PHY. When the PHY has found the necessary edge, it drives the **dfi_wrlvl_resp** signal high, which informs the MC that the procedure is done. The MC then de-asserts the **dfi_wrlvl_en** signal and

issues an MRS command to disable the write leveling logic in the DRAMs. This triggers the PHY to release the **dfi_wrlvl_resp** signal, which completes write leveling.

FIGURE 53. *Write Leveling in PHY Evaluation Mode*



4.10.3.4 PHY Independent Mode

In PHY Independent mode, the PHY is responsible for executing data eye training, gate training or write leveling independent of the MC. In this mode, the associated training interface is not used other than the mode signal to the MC.

The MC should be capable of generating the required MRS commands to enter or exit the test modes of the memory devices. These operations are not automatically generated.

All training sequences, regardless of mode, are expected to be executed after memory initialization. For PHY Independent mode, the update interface may be used to suspend memory commands while the training sequences are executed.

4.11 Low Power Control Handshaking

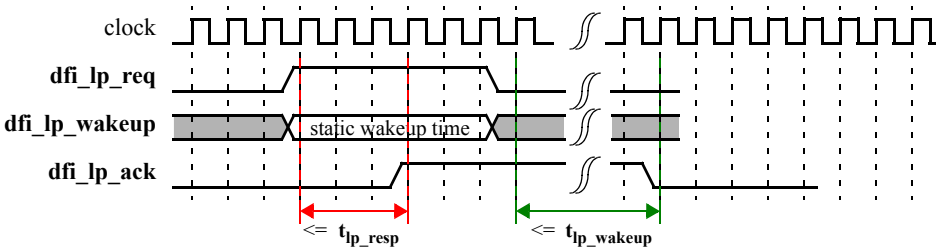
If the PHY has knowledge that the DFI will be idle for a period of time, it may be able to enter a low power state. The DFI contains signals to support a MC-initiated low power opportunity request. The signals used in the low power control interface are: **dfi_lp_req**, **dfi_lp_ack** and **dfi_lp_wakeup**. For more information on these signals, refer to Section 3.7, “Low Power Control Interface”. While using this interface, the DFI bus is expected to remain constant and valid.

Assertion of the **dfi_lp_req** signal indicates to the PHY that the MC is idle. If the request is acknowledged through the assertion of the **dfi_lp_ack** signal, the PHY may enter a low power mode as long as the **dfi_lp_req** signal remains asserted. Once the

dfi_lp_req signal is de-asserted, the PHY must return to normal operating mode within the number of cycles indicated by the **dfi_lp_wakeup** signal. The PHY is not required to change power states, even if a low power opportunity request was acknowledged.

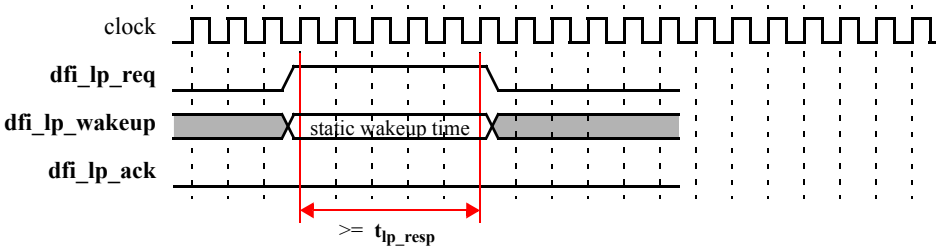
The MC guarantees that **dfi_lp_req** will be asserted and the **dfi_lp_wakeup** signal will be constant for at least $\mathbf{t_{lp\_resp}}$ cycles, allowing the PHY time to respond. The PHY may respond or ignore the low power mode request. To acknowledge the request, the PHY must assert the **dfi_lp_ack** signal within $\mathbf{t_{lp\_resp}}$ clock cycles of the request signal assertion, during which time the MC must hold the **dfi_lp_wakeup** signal constant. Once the request has been acknowledged by the PHY, the MC may de-assert the **dfi_lp_req** signal. The PHY must de-assert the **dfi_lp_ack** signal within $\mathbf{t_{lp\_wakeup}}$ clock cycles after the **dfi_lp_req** signal is de-asserted and be ready for normal operation. Figure 54 shows a sequence in which the request is acknowledged.

FIGURE 54. *Low Power Control Handshaking Timing Diagram*



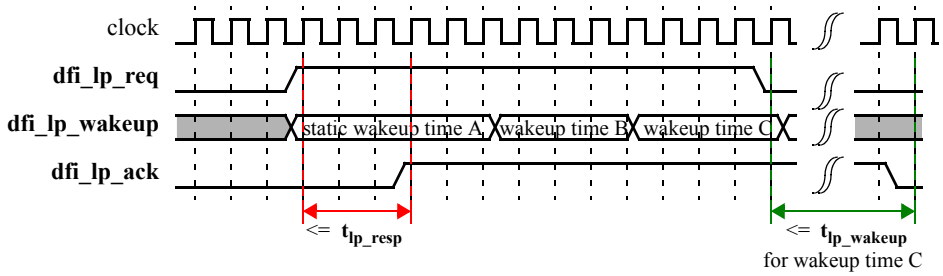
It is important to note that the **dfi_lp_ack** signal is not required to assert when the **dfi_lp_req** signal is asserted. The MC must assert the **dfi_lp_req** signal for at least $\mathbf{t_{lp\_resp}}$ cycles. If the **dfi_lp_ack** signal is not asserted within $\mathbf{t_{lp\_resp}}$ cycles, the PHY must not assert the acknowledge for the current request. The **dfi_lp_req** signal should be de-asserted after $\mathbf{t_{lp\_resp}}$ cycles have elapsed without an acknowledge. This scenario is shown in Figure 55.

FIGURE 55. *Low Power Control Request with No Response*



After the request has been acknowledged, the MC may decide that the PHY has more time to respond than initially defined. The MC is allowed to change the **dfi_lp_wakeup** signal to a larger value as long as both the **dfi_lp_ack** and **dfi_lp_req** signals are asserted. This results in a longer $\mathbf{t_{lp\_wakeup}}$ time for the PHY. The value of the **dfi_lp_wakeup** signal when the **dfi_lp_req** signal is de-asserted will be used to define the $\mathbf{t_{lp\_wakeup}}$ time. Figure 56 shows this situation with the assumption that the wakeup time will be increased with each change.

FIGURE 56. *Low Power Control Handshaking Timing Diagram with Multiple Wakeup Times*



5.0 Signal Timing

The DFI specification does not specify timing values for signaling between the MC and the PHY. The only requirement is that a DFI clock must exist, and all DFI-related signals must be driven by registers referenced to the rising edge of this clock. There are no restrictions on how these signals are received, nor any rules on the source of the DFI clock. Compatibility between the MC and the PHY at given frequencies is dependent on the specification of both the output timing for signals driven and the setup and hold requirements for reception of these signals on the DFI.

However, the DFI signals are categorized into three signal groups that place restrictions on how signals may be driven and captured by DFI devices. All DFI signals may be driven from the DFI clock and captured on the following rising edge of the DFI clock. However, some signals may allow less restrictive timing which may alleviate timing restrictions within the design.

For frequency ratio systems, the PHY must send and receive DFI signals with respect to the rising edge of the DFI clock, even if the signals are driven and captured by registers clocked by the higher frequency DFI PHY clock.

There are three signal groups:

1. **Standard Signals**

   These signals contain timing critical information on a cycle-by-cycle basis and, therefore, must be sent and received on every DFI clock. The Control Interface signals fall into this category because they must be cycle-accurate to properly communicate memory commands.

   Standard signals must be sent on the DFI clock and must be received on every DFI clock period for proper operation. All standard signals are required to meet setup and hold time to the DFI clock at the destination. Neither device should have a dependency on the source clock of the other device; the source and destination clock should always be assumed to be the DFI clock.

2. **State-Retaining Signals**

   These signals contain information that is not single cycle-critical because they retain a state change until either a signal acknowledge is received or a timing parameter has been satisfied. The Update Interface signals fall into this category because all signal state changes are defined in terms of signal responses and timing parameters.

   State-retaining signals may be sent and received in the same way on every DFI clock period. They may also be sent and/or received by a divided frequency clock; provided that the lower frequency clock must be an even multiple (½, ¼, etc.) and phase aligned to the DFI clock.

   All state-retaining signals are required to meet setup and hold time to the DFI clock at the destination regardless of whether the signal is generated from the DFI clock or a lower frequency, phase-aligned clock source. Neither device should have a dependency on the source clock of the other device; the source and destination clock should always be assumed to be the DFI clock. If a lower frequency clock is used, the associated timing parameters must be set to appropriately account for the timing effects of using a lower frequency clock at either the source or destination. The

timing parameters are always defined in terms of the DFI clock regardless of the source and destination clock frequency.

3. **Timer-Based Signals**

   These signals do not have a clock-edge dependency because they are either not required to be valid until a timing parameter has been met or are expected to be static during normal operation (static signals may be changed during idle times). Training signals such as the **dfi_rdlvl_delay_X** and **dfi_rdlvl_resp** are classified as timer-based because they are not valid until the $\mathbf{t_{rdlvl\_load}}$ and $\mathbf{t_{rdlvl\_resplat}}$ timing parameters have been met.

   Timer-based signals do not have to meet setup and hold time to the DFI clock except on the cycle after meeting the associated timing parameter. They may also be changed during idle times in which case the setup and hold times are irrelevant. For the purpose of timing analysis, these signals may be treated as multi-cycle paths.

Each DFI signal is defined into one of these signal groups as shown in Table 16.

TABLE 16. *Signal Group Divisions*

Signal	Signal Group	Associated Timing Parameter [a]
dfi_address (or **dfi_address_pN**)	Standard	N/A
dfi_bank (or **dfi_bank_pN**)	Standard	N/A
dfi_cas_n (or **dfi_cas_pN**)	Standard	N/A
dfi_cke (or **dfi_cke_pN**)	Standard	N/A
dfi_cs_n (or **dfi_cs_pN**)	Standard	N/A
dfi_ctrlupd_ack	State-Retaining	N/A
dfi_ctrlupd_req	State-Retaining	N/A
dfi_data_byte_disable	Timer-Based	static
dfi_dram_clk_disable	Standard	N/A
dfi_freq_ratio	Timer-Based	static
dfi_init_complete	State-Retaining	N/A
dfi_init_start	State-Retaining	N/A
dfi_lp_ack	State-Retaining	N/A
dfi_lp_req	State-Retaining	N/A
dfi_lp_wakeup	State-Retaining	N/A
dfi_odt (or **dfi_odt_pN**)	Standard	N/A
dfi_parity_error	Standard	N/A

TABLE 16. *Signal Group Divisions*

Signal	Signal Group	Associated Timing Parameter [a]
dfi_parity_in	Standard	N/A
dfi_phyupd_ack	State-Retaining	N/A
dfi_phyupd_req	State-Retaining	N/A
dfi_phyupd_type	State-Retaining	N/A
dfi_ras_n (or **dfi_ras_pN**)	Standard	N/A
dfi_rddata (or **dfi_rddata_wN**)	Standard	N/A
dfi_rddata_dnv (or **dfi_rddata_dnv_wN**)	Standard	N/A
dfi_rddata_en (or **dfi_rddata_en_pN**)	Standard	N/A
dfi_rddata_valid (or **dfi_rddata_valid_wN**)	Standard	N/A
dfi_rdlvl_cs_n	Timer-Based	static
dfi_rdlvl_delay_X	Timer-Based	t_{rdlvl_load}
dfi_rdlvl_edge	Timer-Based	static
dfi_rdlvl_en	State-Retaining	N/A
dfi_rdlvl_gate_delay_X	Timer-Based	t_{rdlvl_load}
dfi_rdlvl_gate_en	State-Retaining	N/A
dfi_rdlvl_gate_mode	Timer-Based	static
dfi_rdlvl_gate_req	State-Retaining	N/A
dfi_rdlvl_load	Standard	N/A
dfi_rdlvl_mode	Timer-Based	static
dfi_rdlvl_req	State-Retaining	N/A
dfi_rdlvl_resp	Timer-Based	$t_{rdlvl\_resplat}$
dfi_reset_n (or **dfi_reset_pN**)	Standard	N/A
dfi_we_n (or **dfi_we_pN**)	Standard	N/A
dfi_wrdata (or **dfi_wrdata_pN**)	Standard	N/A
dfi_wrdata_en (or **dfi_wrdata_en_pN**)	Standard	N/A
dfi_wrdata_mask (or **dfi_wrdata_mask_pN**)	Standard	N/A

TABLE 16. *Signal Group Divisions*

Signal	Signal Group	Associated Timing Parameter [a]
dfi_wrlvl_cs_n	Timer-Based	static
dfi_wrlvl_delay_X	Timer-Based	t_{wrlvl_load}
dfi_wrlvl_en	State-Retaining	N/A
dfi_wrlvl_load	Standard	N/A
dfi_wrlvl_mode	Timer-Based	static
dfi_wrlvl_req	State-Retaining	N/A
dfi_wrlvl_resp	Timer-Based	$t_{wrlvl\_resplat}$
dfi_wrlvl_strobe	Standard	N/A

a. Signals falling in the Standard and State-Retaining Signal Groups have no timing parameter correlations.

6.0 Glossary

TABLE 17. *Glossary of Terms*

Term	Definition
MC	DDR Memory Controller logic
PHY	DDR Physical Interface logic
Idle	The DFI bus is considered idle when the control interface is not sending any commands and all read and write data has reached its destination (memory or MC).
DFI clock frequency	Defines the clock frequency of the MC.
PHY DFI clock frequency	Defines the clock frequency of the PHY. For matched systems, this will be the same as the DFI clock frequency. For frequency ratio systems, the PHY DFI clock frequency must be two or four times the DFI clock frequency. These clocks must be aligned in phase.
DFI data word	One phase of read data. A DFI data word is twice the width of the memory data bus and corresponds to a single memory word transfer across the DFI bus.
DFI Address Width	The width of the address bus on the DFI interface. This is generally the same width as the DRAM address bus.
DFI Bank Width	The number of bank bits on the DFI interface. This is generally the same number of bits as the number of bank pins on the DRAM device.
DFI Control Width	The number of bits required to control the memory devices, usually a single bit.
DFI Chip Select Width	The number of chip select bits on the DFI interface. This is generally the same number of bits as the number of chip select pins on the DRAM device.
DFI Data Width	The width of the datapath on the DFI interface. This is generally twice the DRAM data width.
DFI Data Enable Width	The width of the datapath enable signals on the DFI interface. For PHYs with an 8-bit slice, this will generally be 1/16th of the DFI Data Width to provide a single enable bit per memory data slice, but may be 1/4, 1/8, 1/32, or any other ratio.
DFI Read Data Valid Width	The width of the datapath valid signals on the DFI interface. For PHYs with an 8-bit slice, this will generally be 1/16th of the DFI Data Width to provide a single valid bit per memory data slice, but may be 1/4, 1/8, 1/32, or any other ratio. All bits of the signal must hold the same value.
DFI Read Leveling Delay Width	The number of bits required to communicate read delay information to the PHY.
DFI Read Leveling Gate Delay Width	The number of bits required to communicate gate training delay information to the PHY.
DFI Read Leveling MC IF Width	The number of bits used to control the read leveling interface from the MC perspective. The MC Read Leveling signals are generally fanned out such that a copy of the signal can be sent to each PHY memory data slice.
DFI Read Leveling PHY IF Width	The number of bits used to control the read leveling interface from the PHY perspective. The PHY may drive a signal from each memory data slice or combine the signals into a single signal.
DFI Read Leveling Response Width	The number of bits used to communicate read leveling status to the MC. The PHY Read Leveling response may be one bit per memory data slice or one bit per bit on the memory data bus. If this width is the same width as the memory data bus, gate training information should be returned on the lowest bit of each data slice.
DFI Write Leveling Delay Width	The number of bits required to communicate write delay information to the PHY.

TABLE 17. *Glossary of Terms*

Term	Definition
DFI Write Leveling MC IF Width	The number of bits used to control the write leveling interface from the MC perspective. The MC Write Leveling signals are generally fanned out such that a copy of the signal can be sent to each PHY memory data slice.
DFI Write Leveling PHY IF Width	The number of bits used to control the write leveling interface from the PHY perspective. The PHY may drive a signal from each memory data slice or combine the signals into a single signal.
DFI Write Leveling Response Width	The number of bits used to communicate write leveling status to the MC. The PHY should drive a single bit per memory data slice.

7.0 DFI Device Spec Sheet

TABLE 18. *DFI Settings Specifications*

Terms	Relevant Specifications	MC Settings		PHY Settings	
		Max	Min	Max	Min
DFI clock frequency	2.0 and 2.1				
PHY DFI clock frequency	2.1				
DFI Address Width	1.0, 2.0 and 2.1				
DFI Bank Width	1.0, 2.0 and 2.1				
DFI Control Width	1.0, 2.0 and 2.1				
DFI Chip Select Width	1.0, 2.0 and 2.1				
DFI Data Width	1.0, 2.0 and 2.1				
DFI Data Enable Width	2.0 and 2.1				
DFI Read Data Valid Width	2.0 and 2.1				
DFI Read Leveling Delay Width	2.0 and 2.1				
DFI Read Leveling Gate Delay Width	2.0 and 2.1				
DFI Read Leveling MC IF Width	2.0 and 2.1				
DFI Read Leveling PHY IF Width	2.0 and 2.1				
DFI Read Leveling Response Width	2.0 and 2.1				
DFI Write Leveling Delay Width	2.0 and 2.1				
DFI Write Leveling MC IF Width	2.0 and 2.1				
DFI Write Leveling PHY IF Width	2.0 and 2.1				
DFI Write Leveling Response Width	2.0 and 2.1				

TABLE 19. *Timing Parameter Settings*

Parameter	Relevant Specifications	Spec Requirement		Device Specification	
		Max	Min	Max	Min
t_{ctrl_delay}	1.0, 2.0 and 2.1	0	- [a]		
t_{phy_wrdata}	2.1	0	- [a]		
$t_{phy\_wrdelay}$	2.1	0	1 or 3 [b]		
t_{phy_wrlat}	1.0, 2.0 and 2.1	0	- [a]		
t_{phy_rdlat}	1.0, 2.0 and 2.1	0	- [a]		
t_{rddata_en}	1.0, 2.0 and 2.1	0	- [a]		
$t_{ctrlupd\_interval}$	2.0 and 2.1	- [c]	- [a]		

TABLE 19. *Timing Parameter Settings*

Parameter	Relevant Specifications	Spec Requirement		Device Specification	
		Max	Min	Max	Min
$t_{ctrlupd\_min}$	1.0, 2.0 and 2.1	1	- [a]		
$t_{ctrlupd\_max}$	1.0, 2.0 and 2.1	- [c]	- [a]		
t_{phyupd_type0}	1.0, 2.0 and 2.1	1	- [a]		
t_{phyupd_type1}	1.0, 2.0 and 2.1	1	- [a]		
t_{phyupd_type2}	1.0, 2.0 and 2.1	1	- [a]		
t_{phyupd_type3}	1.0, 2.0 and 2.1	1	- [a]		
t_{phyupd_resp}	1.0, 2.0 and 2.1	1	- [a]		
$t_{dram\_clk\_disable}$	1.0, 2.0 and 2.1	0	- [a]		
$t_{dram\_clk\_enable}$	1.0, 2.0 and 2.1	0	- [a]		
$t_{init\_complete}$	2.1	0	- [a]		
t_{init_start} (optional)	2.1	0	- [a]		
$t_{phy\_paritylat}$ (optional)	2.1	4	- [a]		
t_{rdlvl_dll}	2.0 and 2.1	1	- [a]		
t_{rdlvl_en}	2.0 and 2.1	1	- [a]		
t_{rdlvl_load}	2.1	1	- [a]		
t_{rdlvl_max}	2.0 and 2.1	- [c]	- [a]		
t_{rdlvl_resp}	2.0 and 2.1	1	- [a]		
$t_{rdlvl\_resplat}$	2.0 and 2.1	1	- [a]		
t_{rdlvl_rr}	2.0 and 2.1	- [c]	- [a]		
t_{wrlvl_dll}	2.0 and 2.1	1	- [a]		
t_{wrlvl_en}	2.0 and 2.1	1	- [a]		
t_{wrlvl_load}	2.1	1	- [a]		
t_{wrlvl_max}	2.0 and 2.1	- [c]	- [a]		
t_{wrlvl_resp}	2.0 and 2.1	1	- [a]		
$t_{wrlvl\_resplat}$	2.0 and 2.1	1	- [a]		
t_{wrlvl_ww}	2.0 and 2.1	- [c]	- [a]		
t_{lp_resp}	2.1	1 [d]	7		
t_{lp_wakeup}	2.1	0	- [e]		

a. The DFI does not specify a maximum value. The range of values supported is implementation-specific.

b. The PHY must support a maximum write delay of one less than the clock ratio. A 1:2 frequency ratio PHY must support values of 0 and 1. A 1:4 frequency ratio PHY must support a range of 0-3.
c. The DFI does not specify a minimum value. The range of values supported is an implementation-specific design parameter.
d. It is recommended to fix this timing parameter at 7 cycles.
e. There is no maximum value defined for this timing parameter.