DDR PHY Interface (DFI) Specification

Version 1.0

DENALI SOFTWARE, INC.
1850B Embarcadero Rd.
Palo Alto, CA 94303
Tel: (650) 461-7200
Fax: (650) 461-7209

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

Denali Software, Inc. Palo Alto, CA 94303

Release Information

Revision Number	Date	Change
1.0	30 Jan 2007	Initial Release

Proprietary Notice

RESTRICTED RIGHTS LEGEND

Destination Control Statement

Trademarks

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	Samsung
Synopsys	

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 and DDR2 memory devices. The protocol defines the signals, signal relationships, and timing parameters required to transfer control information, read and write 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 5.0, “Glossary”.

2.0 Architecture

All signals defined by the DFI are required to be driven by registers clocked on the rising edge of the DFI clock. The DFI specification places no restrictions on how these signals are received, nor does it dictate the source of the DFI clock. The only requirement is that the DFI clock must exist and all DFI-related signals must be referenced from this 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.

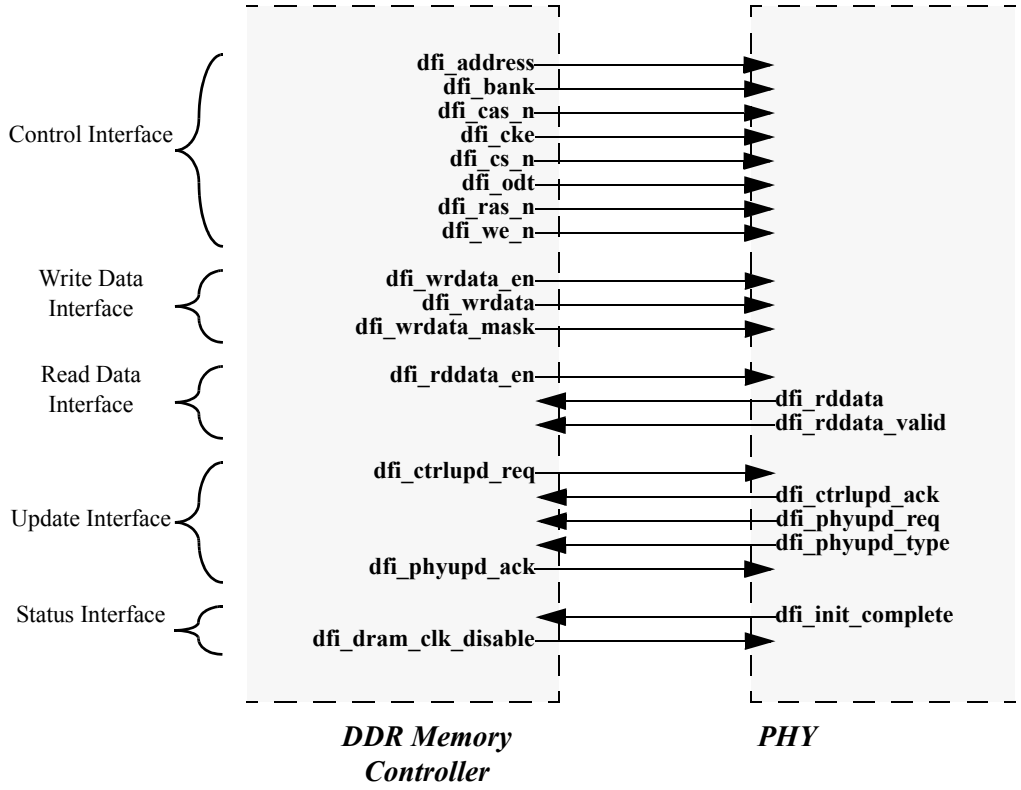
The DFI specification includes signal and timing parameter descriptions required for DFI compliance. DFI compatibility is dependent on the widths and values of signals and timing parameters provided by the MC and the PHY. Fully compliant DFI devices may be incompatible if their DFI signal widths and/or their timing parameters are inconsistent.

Fully compliant DFI devices 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.

The DFI specification allows certain timing parameters to be specified as fixed values or as constants based on other values in the system. However, these timing parameters are expected to be constant during DFI operation.

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

The Control Interface is a reflection of the DRAM control interface for address, bank, chip select, row strobe, column strobe, write enable, clock enable and ODT control. The Write Data Interface and Read Data Interface are used to pass 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 as well as to control the presence of valid clocks to the DRAM interface.

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. This parameter, along with the $\mathbf{t_{phy\_wrlat}}$ timing parameter, are used to align the command and the write data on the DRAM interface. Refer to Table 4, "Write Data Timing Parameter" for more information on $\mathbf{t_{phy\_wrlat}}$.

The DFI specification supports DRAM devices operating at a multiple of the frequency of the DFI. In this case, the DFI address and control signal widths are extended in accordance with the frequency multiple being used. For example, if the DRAM frequency is twice the DFI frequency, then the **dfi_ras_n** signal would be 2 bits wide (instead of 1 for matched DFI/DRAM frequency) and the **dfi_address** signal would be twice as wide as in the matched frequency case. The control signals on the DFI must follow a little endian convention sending lower bits first followed by higher bits.

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 1 and Table 2.

TABLE 1. *Control Signals*

Signal	From	Width	Default	Description
dfi_address	MC	DFI Address Width	N/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** signal when reflecting this data to the DRAM devices.
dfi_bank	MC	DFI Bank Width	N/A	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** signal when reflecting this data to the DRAM devices.
dfi_cas_n	MC	DFI Control Width	0x1	DFI column address strobe. These signal(s) define the CAS information that is intended for the DRAM devices for all control commands.
dfi_cke	MC	DFI Chip Select Width	0x0	DFI clock enable. These signal(s) define the CKE information that is intended for the DRAM devices for all control commands.
dfi_cs_n	MC	DFI Chip Select Width	0x1	DFI chip selects. These signal(s) define the CS information that is intended for the DRAM devices for all control commands.
dfi_odt	MC	DFI Chip Select Width	0x0	DFI on-die termination control signal. These signal(s) define the ODT information that is intended for the DRAM devices for all control commands.

TABLE 1. *Control Signals*

Signal	From	Width	Default	Description
dfi_ras_n	MC	DFI Control Width	0x1	DFI row address strobe. These signal(s) define the RAS information that is intended for the DRAM devices for all control commands.
dfi_we_n	MC	DFI Control Width	0x1	DFI write enable. These signal(s) define the WEN information that is intended for the DRAM devices for all control commands.

TABLE 2. *Control Timing Parameters*

Parameter	Defined By	Min	Max	Unit	Description
$\mathbf{t_{ctrl\_delay}}$	PHY	0	-[a]	Cycles	Specifies the number of DFI clocks after an assertion or de-assertion of the DFI control signals that the control signals at the PHY-DRAM reflect the assertion or de-assertion.

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

3.2 Write Data Interface

The write data interface handles transmitting 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 parameter $\mathbf{t_{phy\_wrlat}}$ are described in Table 3 and Table 4.

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 one cycle after the **dfi_wrdata_en** signal is asserted. The **dfi_wrdata_mask** signal follows the same timing as the **dfi_wrdata** signal, one cycle 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. Once this value has been established, it must remain constant during the operation of the system and the **dfi_wrdata_en** signal must be asserted based on this timing parameter.

More information on the write data interface is provided in Section 4.3, “Write Transactions”.

TABLE 3. *Write Data Signals*

Signal	From	Width	Default	Description
dfi_wrdata_en	MC	DFI Data Width / 8	0x0	Write data and data mask valid signals. These signals must be asserted one cycle 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. There is a single enable signal bit for each byte of the **dfi_wrdata** bus. The **dfi_wrdata_en**[0] signal corresponds to the **dfi_wrdata**[7:0] signals, the **dfi_wrdata_en**[1] signal corresponds to **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_en** signal corresponds to the most significant partial byte of data.
dfi_wrdata	MC	DFI Data Width	N/A	Write data signal. The write data stream must begin one cycle after the **dfi_wrdata_en** signal is asserted for the number of cycles that the **dfi_wrdata_en** signal is asserted.
dfi_wrdata_mask	MC	DFI Data Width / 8	N/A	Write data byte mask signal. 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.

TABLE 4. *Write Data Timing Parameter*

Parameter	Defined By	Min	Max	Unit	Description
$\mathbf{t_{phy\_wrlat}}$	PHY	0	-[a]	Cycles	Specifies the number of DFI clocks between when a write command is sent on the DFI control interface (**dfi_cas_n** = 0, **dfi_ras_n** = 1, **dfi_we_n** = 0, **dfi_cs_n** = active) 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 max value. The range of values supported is implementation-specific.

3.3 Read Data Interface

The read data interface handles returning 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** along with the related timing parameters $\mathbf{t_{rddata\_en}}$ and $\mathbf{t_{phy\_rdlat}}$ are described in Table 5 and Table 6.

The **dfi_rddata_en** signal is asserted $\mathbf{t_{rddata\_en}}$ 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. Multiple read commands can be asserted on the DFI interface while the **dfi_rddata_en** signal is asserted. 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 requirement 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 number of cycles 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. As in the case for the $\mathbf{t_{rddata\_en}}$ parameter, once this value has been established, it must remain constant during the operation of the system. The two timing parameters $\mathbf{t_{rddata\_en}}$ and $\mathbf{t_{phy\_rdlat}}$ work together to define a constant 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.

More information on the read data interface is provided in Section 4.4, “Read Transactions”.

TABLE 5. *Read Data Signals*

Signal	From	Width	Default	Description
dfi_rddata_en	MC	DFI Data Width / 8	0x0	Read data enable signal. The **dfi_rddata_en** signal must be asserted $\mathbf{t_{rddata\_en}}$ 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. There is a single enable signal bit for each byte of the DRAM data specified by the DFI Data Width. If the **dfi_rddata** bus is not a multiple of 8, then the uppermost bit of the **dfi_rddata_en** signal corresponds to the most significant partial byte of data.
dfi_rddata	PHY	DFI Data Width	N/A	Read data signal. Read data is expected to be received at the MC $\mathbf{t_{phy\_rdlat}}$ cycles after the **dfi_rddata_en** signal is asserted.
dfi_rddata_valid	PHY	1 bit	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.

TABLE 6. *Read Data Timing Parameters*

Parameter	Defined By	Min	Max	Unit	Description
$\mathbf{t_{phy\_rdlat}}$	PHY	0	-[a]	Cycles	Specifies the time from the assertion of the **dfi_rddata_en** signal to the assertion of the **dfi_rddata_valid** signal.
$\mathbf{t_{rddata\_en}}$	System	0	-[a]	Cycles	Specifies the time from the assertion of a read command on the DFI (**dfi_ras_n** = 1, **dfi_cas_n** = 0, **dfi_we_n** = 1, **dfi_cs_n** = active) 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 max value. The range of values supported is implementation-specific.

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 modes where the DFI read, write, and control interface are suspended. 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. 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 does not force the MC to issue update requests nor does it specify an interval in which requests must be offered. If the MC chooses to offer update requests, it must follow the specified protocol.

If a PHY initiates an update request by asserting the **dfi_phyupd_req** signal, the request must be acknowledged 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. The MC is responsible for placing the system in a state where the DRAMs can operate without communication across the DFI for the number of cycles indicated by the **dfi_phyupd_type** signal and the associated $\mathbf{t_{phyupd\_typeX}}$ timing parameter. Refer to Section 4.5.2, “PHY-Initiated Update” for more details on this protocol.

The signals and timing parameters in the update interface are listed in Table 7 and Table 8.

TABLE 7. *Update Interface Signals*

Signal	From	Width	Default	Description
dfi_ctrlupd_ack	PHY	1 bit	0x0	The **dfi_ctrlupd_ack** signal is asserted to acknowledge a MC-initiated update request. The PHY is not required to acknowledge this request. 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 available for at least $\mathbf{t_{ctrlupd\_min}}$ cycles.
dfi_ctrlupd_req	MC	1 bit	0x0	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 if the PHY chooses not to perform an update at this time. 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. • 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 7. *Update Interface Signals*

Signal	From	Width	Default	Description
dfi_phyupd_ack	MC	1 bit	0x0	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 **dfi_phyupd_ack** signal must assert 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. 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	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 de-assertion of the **dfi_phyupd_req** signal triggers the de-assertion of the **dfi_phyupd_ack** signal.
dfi_phyupd_type	PHY	2 bits	N/A	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.

TABLE 8. *Update Timing Parameters*

Parameter	Defined By	Min	Max	Unit	Description
$\mathbf{t_{ctrlupd\_min}}$	MC	1	-[a]	Cycles	Specifies the minimum number of DFI clock cycles that the **dfi_ctrlupd_req** signal must be asserted.
$\mathbf{t_{ctrlupd\_max}}$	MC	-[b]	-[a]	Cycles	Specifies the maximum number of DFI clock cycles that the **dfi_ctrlupd_req** signal can assert.

TABLE 8. *Update Timing Parameters*

Parameter	Defined By	Min	Max	Unit	Description
t_{phyupd_type0}	PHY	1	-[a]	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	-[a]	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	-[a]	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	-[a]	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	-[a]	Cycles	Specifies the maximum number of 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 max 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.

3.5 Status Interface

The DFI requires status information for initialization and clock control to the DRAM devices. More information on the clock disable interface is provided in Section 4.6,

"DFI Clock Disabling". The signals and timing parameters for the status interface are listed in Table 9 and Table 10.

TABLE 9. *Status Interface Signals*

Signal	From	Width	Default	Description
dfi_dram_clk_disable	MC	DFI Chip Select Width	0x0	DRAM clock disable signal. 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_init_complete	PHY	1 bit	0x0	PHY initialization complete signal. The **dfi_init_complete** signal indicates that the PHY is able to respond to any proper stimulus on the DFI. All other DFI signals must be held at their default values until the **dfi_init_complete** signal asserts. De-assertion of the **dfi_init_complete** signal is not permitted by the DFI specification unless a system reset is performed.

TABLE 10. *Status Timing Parameters*

Parameter	Defined By	Min	Max	Unit	Description
$t_{dram\_clk\_disable}$	PHY	0	-[a]	Cycles	Specifies the number of clocks 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. 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]	Cycles	Specifies the number of clocks 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. 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 max value. The range of values supported is an implementation-specific design parameter.

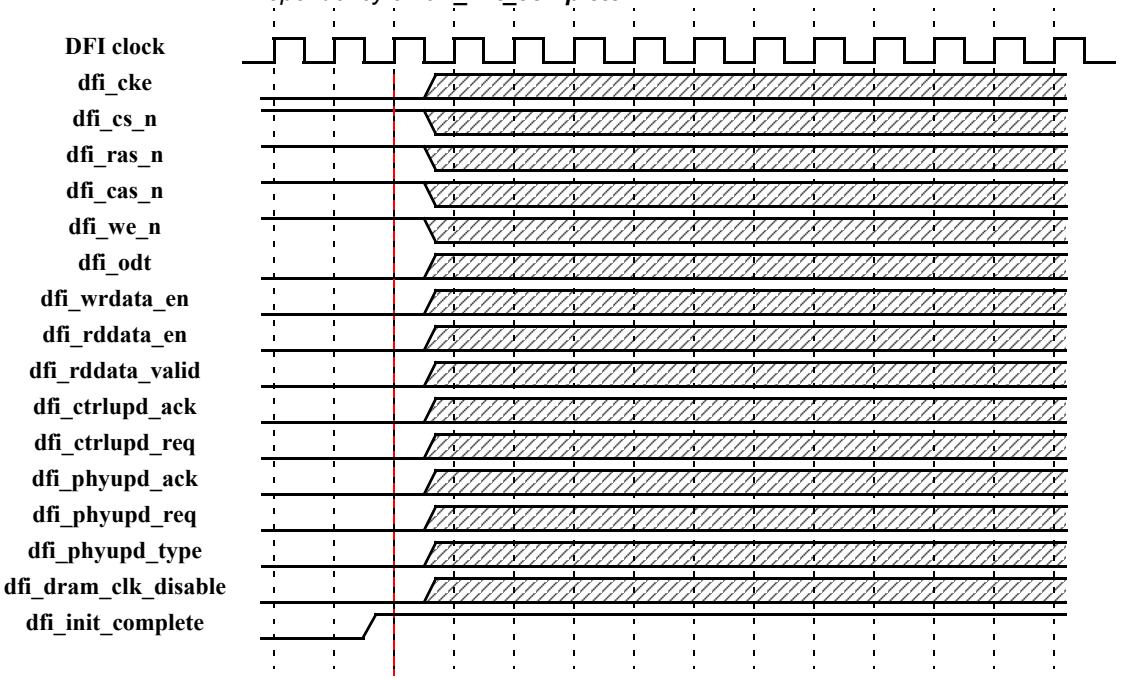
4.0 Functional Use

4.1 Initialization

The DFI specification requires that, as long as the **dfi_init_complete** signal is not asserted, the DFI signals must remain at default value. As shown in Figure 2, "Depen-

dency on dfi_init_complete", once the **dfi_init_complete** signal is asserted, all other DFI signals are able to assert in accordance with the DFI specification.

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



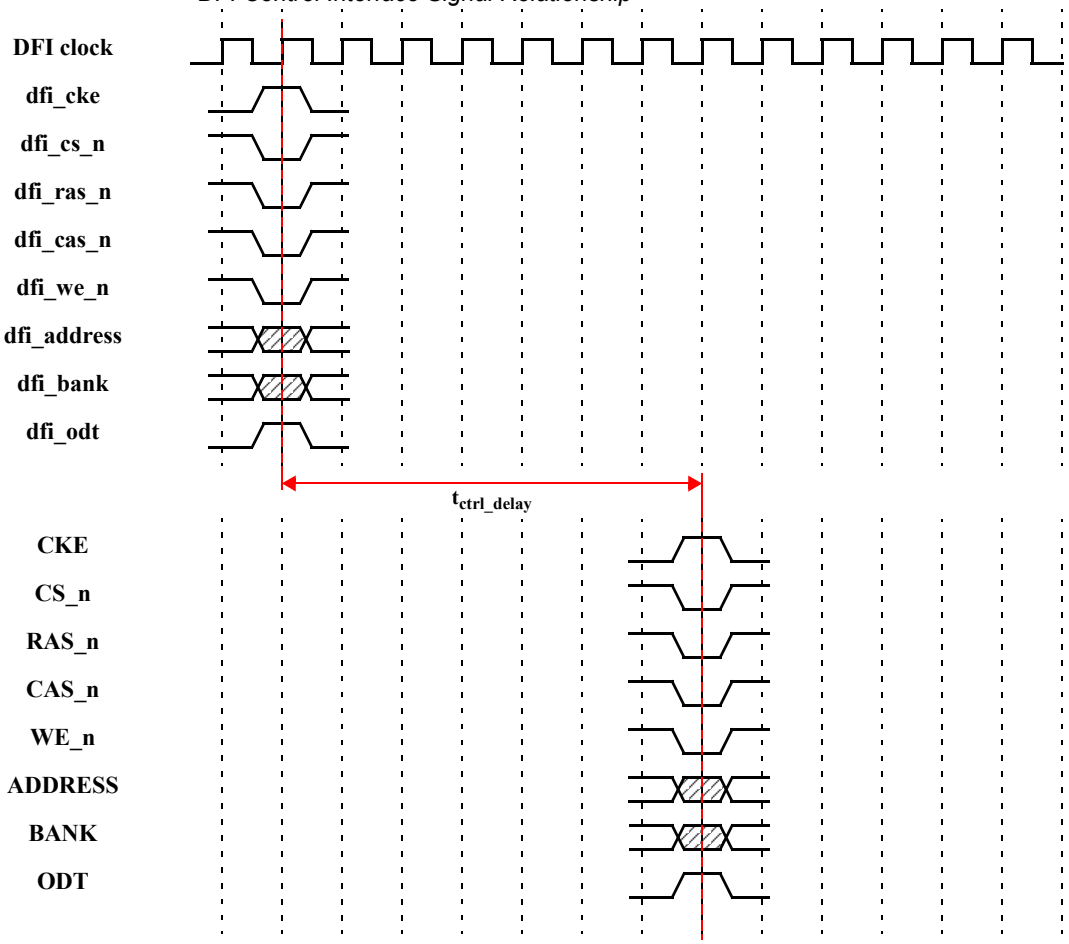
The DFI specification does not impose or dictate a reset sequence for either the PHY or the MC. However, the assertion of the **dfi_init_complete** signal signifies that the PHY is ready to respond to any assertions on the DFI by the MC. This does not ensure data integrity to the DRAMs, only that the 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 data eye training prior to signal assertion.

4.2 Control Signals

The DFI control signals **dfi_address**, **dfi_bank**, **dfi_cas_n**, **dfi_cke**, **dfi_cs_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. This latency is defined through the timing parameter $\mathbf{t_{ctrl\_delay}}$. Refer to Figure 3, "DFI Control Interface Signal Relationship" for a graphical representation.

FIGURE 3. *DFI Control Interface Signal Relationship*



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.

The control signals are defined as vectors to allow for PHY and MC systems where the relative clocking of the DFI and the DRAM interface are not source-synchronous. As an example, if the DRAM interface runs at twice the frequency of the DFI interface, then the **dfi_address** signal will be twice as many bits as the DRAM address pins.

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 $\mathbf{t_{phy\_wrlat}}$ parameter. 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 the **dfi_wrdata_en** signal must be asserted for the number of cycles required to complete the write transaction(s) 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**) is valid one cycle after the assertion of the **dfi_wrdata_en** signal on the DFI. The **dfi_wrdata_en** signal must de-assert on the cycle before the last valid data is transferred on the **dfi_wrdata** bus.

Four situations are presented in Figure 4, Figure 5, Figure 6, and Figure 7. All four situations show system behavior with two write transactions.

Figure 4, shows back-to-back writes for a system with a $\mathbf{t_{phy\_wrlat}}$ of zero. The **dfi_wrdata_en** signal is asserted with the write command for this situation, and is asserted for two cycles 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 4. *Back-to-Back Contiguous Writes (DDR1 Example)*

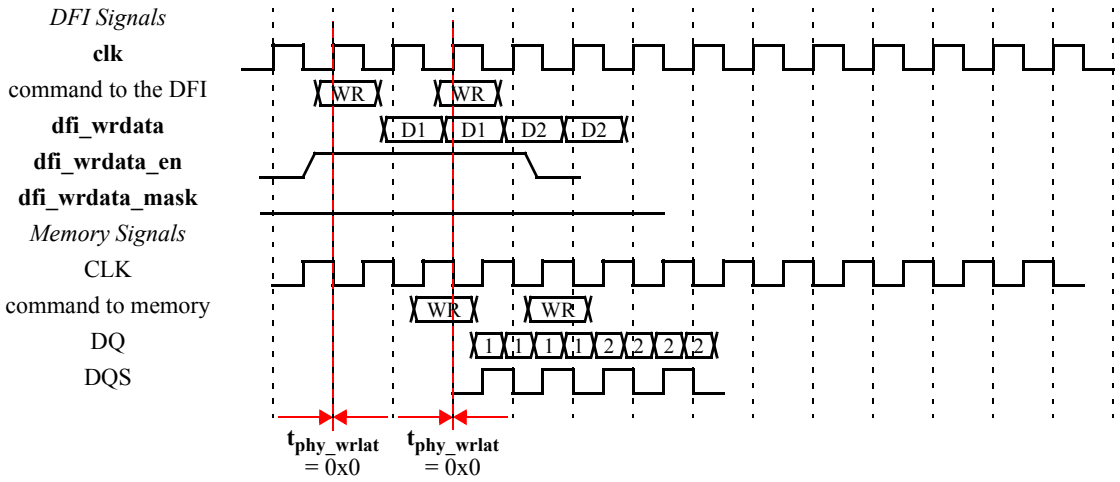


Figure 5 shows an interrupted write command. The **dfi_wrdata_en** signal must be asserted for 4 cycles for each of these write transactions. However, since the first write is interrupted, the **dfi_wrdata_en** signal is asserted for a portion of the first transaction and the complete second transaction. 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 5. *Back-to-Back Interrupted Contiguous Writes (DDR2 Example)*

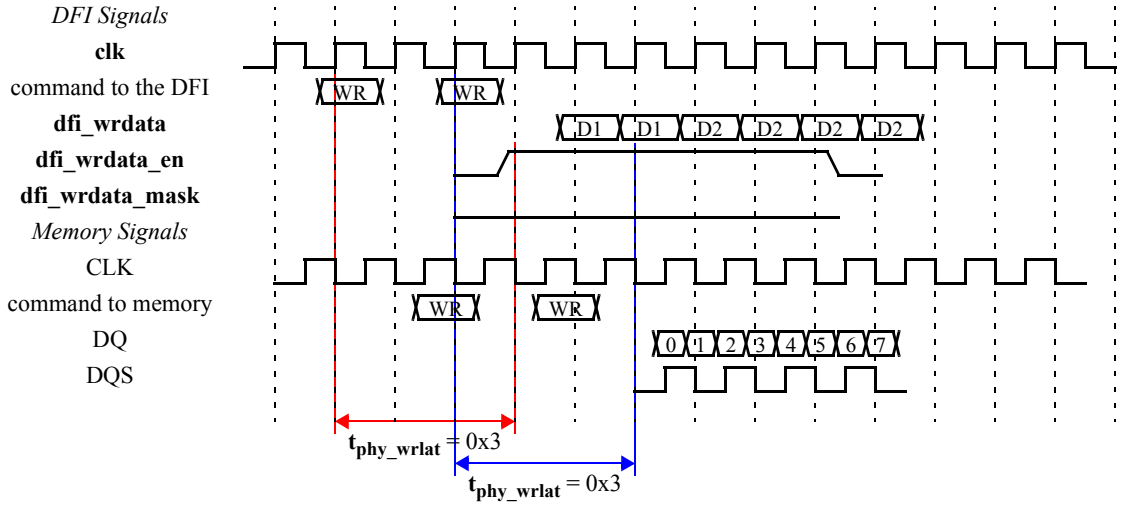


Figure 6 and Figure 7 also show two complete write commands, with different $\mathbf{t_{phy\_wrlat}}$ timing parameters. The **dfi_wrdata_en** signal will be asserted for two cycles for each write transaction. In Figure 6, some portion of the write data is masked by asserting the **dfi_wrdata_mask** signal. 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 6. *Two Independent Writes (DDR1 Example)*

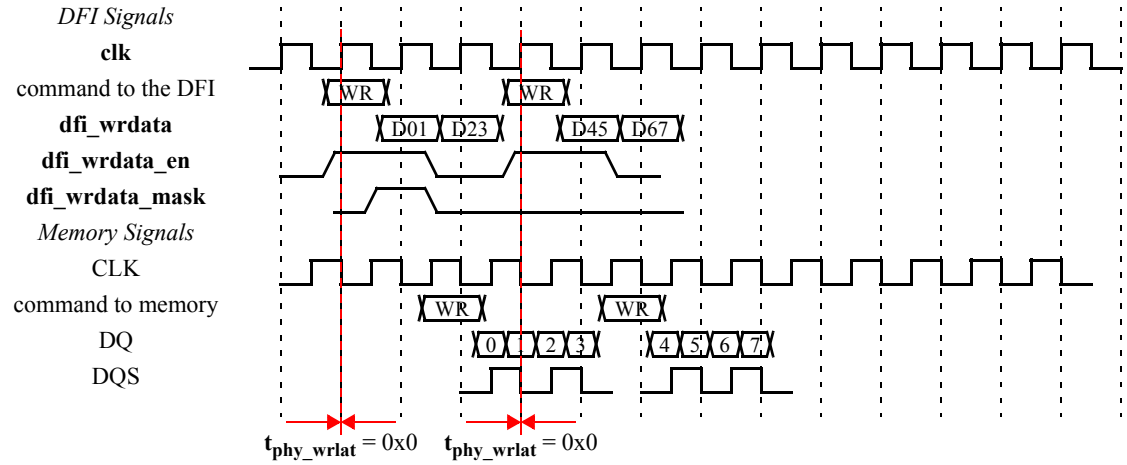
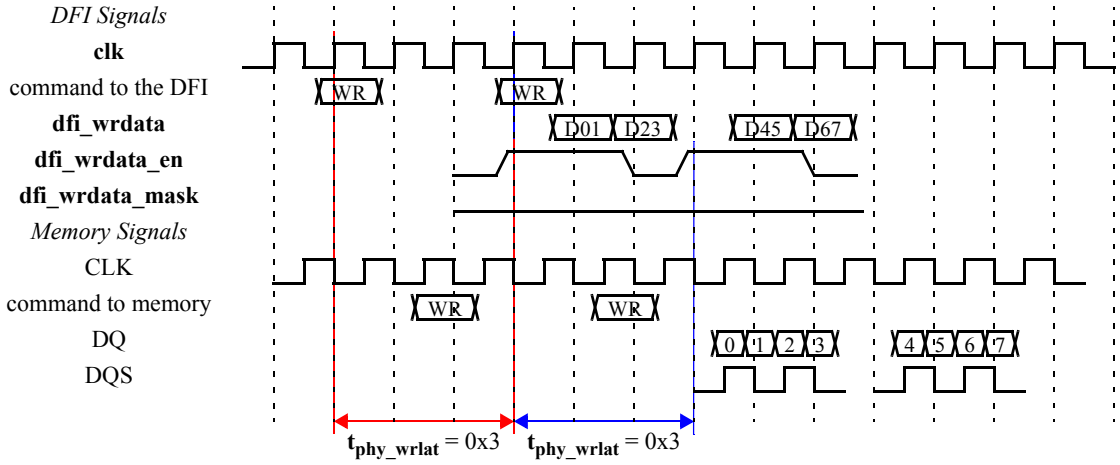


FIGURE 7. *Two Independent Writes (DDR2 Example)*



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**) 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 timing from when a read command is sent on the DFI to when the read data is returned to the DFI must be a constant for a particular application environment. The exact time can be adjusted as long as both the MC and the PHY coordinate the change such that the DFI specification is still maintained. There are two timing parameters that generate this constant: $\mathbf{t_{rddata\_en}}$ and $\mathbf{t_{phy\_rdlat}}$. 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, $\mathbf{t_{phy\_rdlat}}$ cycles after the **dfi_rddata_en** signal for that command is asserted.

Four situations are presented in Figure 8, Figure 9, Figure 10, and Figure 11. Figure 8, 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 8. *Single Read Transaction of 2 Data Words*

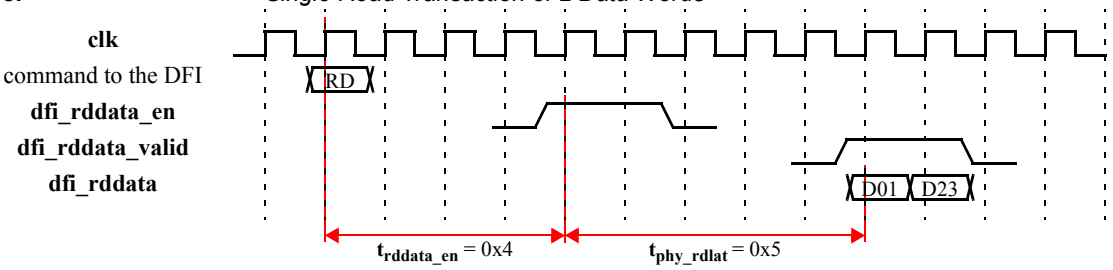


Figure 9 shows an interrupted read command. The **dfi_rddata_en** signal must be asserted for 4 cycles for each of these read transactions. However, since the first read is interrupted, the **dfi_rddata_en** signal is asserted for a portion of the first transaction and the complete second transaction. The **dfi_rddata_en** signal will not de-assert between read commands.

FIGURE 9. *Back-to-Back Read Transactions with First Read Burst Interrupted (DDR1 Example BL=8)*

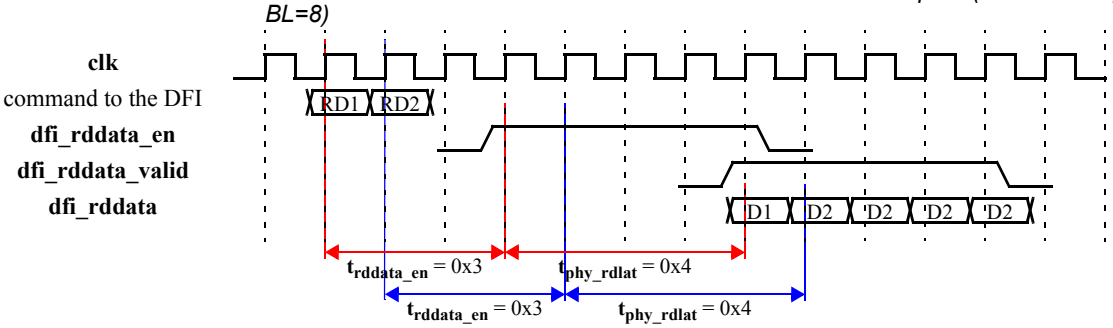
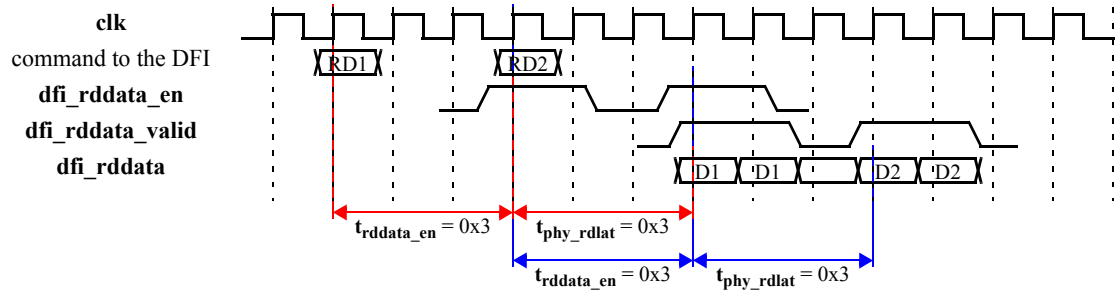


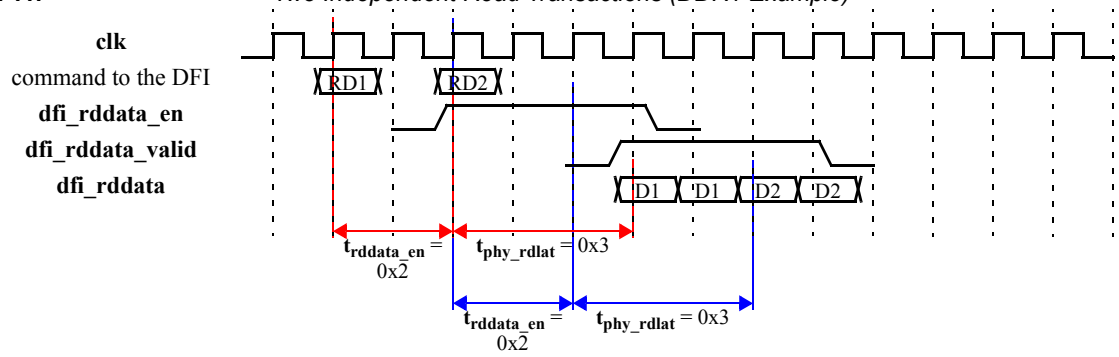
Figure 10 and Figure 11 also show two complete read transactions. The **dfi_rddata_en** signal will be asserted for two cycles for each read transaction. In Figure 10, the $\mathbf{t_{rddata\_en}}$ timing and the timing between the read commands causes the **dfi_rddata_en** signal to be de-asserted between 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 10. *Two Independent Read Transactions (DDR2 Example)*



In Figure 11, 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 11. *Two Independent Read Transactions (DDR1 Example)*



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

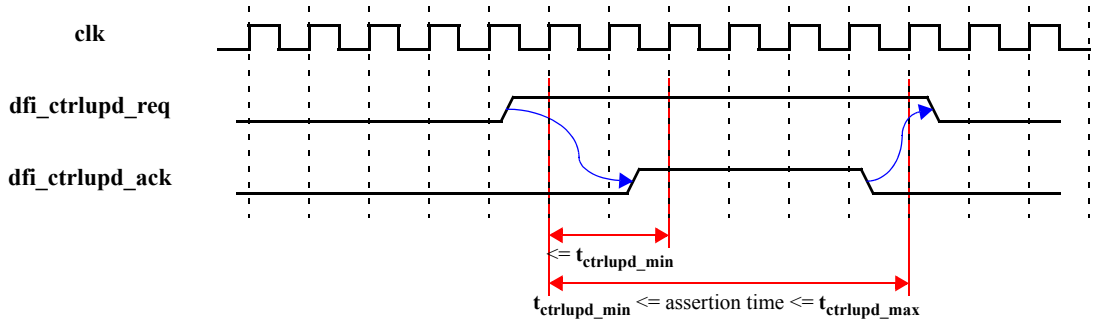
During normal operation, the MC may encounter idle time during which no commands are being issued to the memory devices and all outstanding read and write data have have been transferred on the DFI. Assertion of the **dfi_ctrlupd_req** signal indicates the control, read and write interfaces on the DFI are idle.

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.

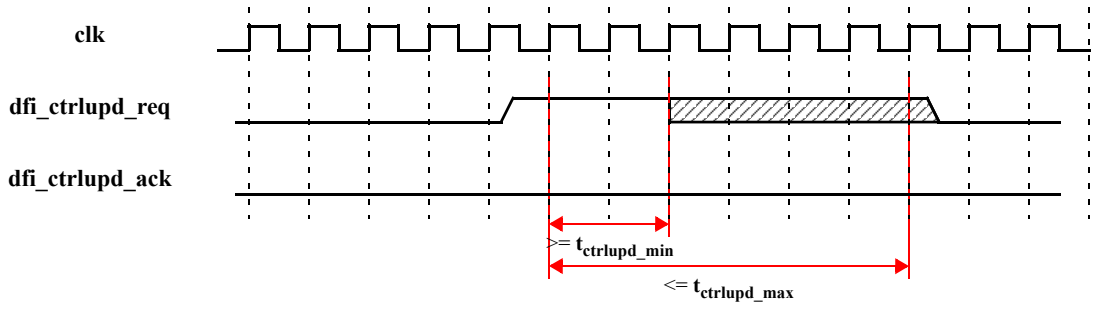
The MC must hold the **dfi_ctrlupd_req** signal as long as the **dfi_ctrlupd_ack** signal is asserted, and must de-assert the **dfi_ctrlupd_req** signal before $\mathbf{t_{ctrlupd\_max}}$ expires. Note that the number of cycles after the **dfi_ctrlupd_ack** signal de-asserts before the **dfi_ctrlupd_req**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

FIGURE 12. *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 $\mathbf{t_{crtlupd\_min}}$, but the total number of cycles that the **dfi_ctrlupd_req** signal is asserted must not exceed $\mathbf{t_{ctrlupd\_max}}$.

FIGURE 13. *MC-Initiated Update with No Response*



4.5.2 PHY-Initiated Update

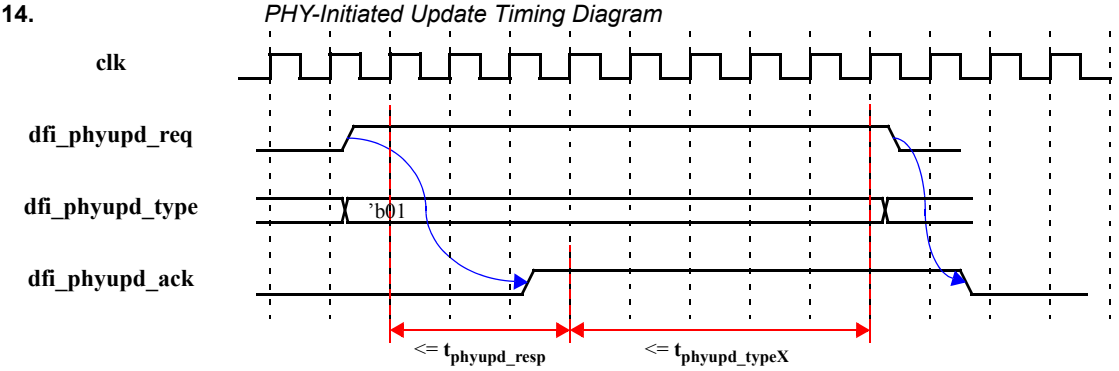
The PHY may also trigger the DFI into an idle state. This update process utilizes three signals: **dfi_phyupd_req**, **dfi_phyupd_type** and **dfi_phyupd_ack**. The **dfi_phyupd_req** signal indicates the need for idle time on the DFI, the **dfi_phyupd_type** signal defines the type of update required, 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.

Unlike MC-initiated updates, the MC must respond to a PHY update request.

FIGURE 14. *PHY-Initiated Update Timing Diagram*

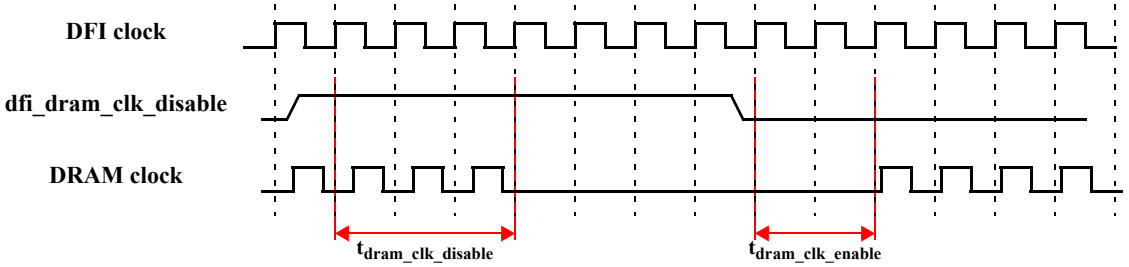


4.6 DFI Clock Disabling

The DFI contains a **dfi_dram_clk_disable** signal which controls the DRAM clock signal to the DRAM device(s). In the default mode, 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.

FIGURE 15. *DRAM Clock Disable Behavior*



5.0 Glossary

TABLE 11. *Glossary of Terms*

Term	Definition
DFI Data Width	The width of the datapath on the DFI interface. For matched frequency PHYs, this is generally twice the DRAM data width and for double-frequency PHYs, this is generally four times the width of the DRAM data width.
DFI Address Width	The width of the address bus on the DFI interface. For matched frequency PHYs, this is generally the same width as the DRAM address bus. For higher-frequency PHYs, this is generally a multiple of the DRAM address bus.
DFI Bank Width	The number of bank bits on the DFI interface. For matched frequency PHYs, this is generally the same number of bits as the number of bank pins on the DRAM device. For higher-frequency PHYs, this is generally a multiple of the number of DRAM bank pins.
DFI Control Width	The number of bits required to control the memory devices. For matched frequency PHYs, a single bit will be sufficient. For double-frequency PHYs, each of these signals will be replicated to multi-bit signals.
DFI Chip Select Width	The number of chip select bits on the DFI interface. For matched frequency PHYs, this is generally the same number of bits as the number of chip select pins on the DRAM device. For higher-frequency PHYs, this is generally a multiple of the number of DRAM chip select pins.
MC	DDR Memory Controller logic
PHY	DDR Physical Interface logic