17. Timing Diagrams for ALTMEMPHY IP

EMI_RM_015-1.3

This chapter shows timing diagrams for the DDR, DDR2, and DDR3 SDRAM high-performance controllers II (HPC II).

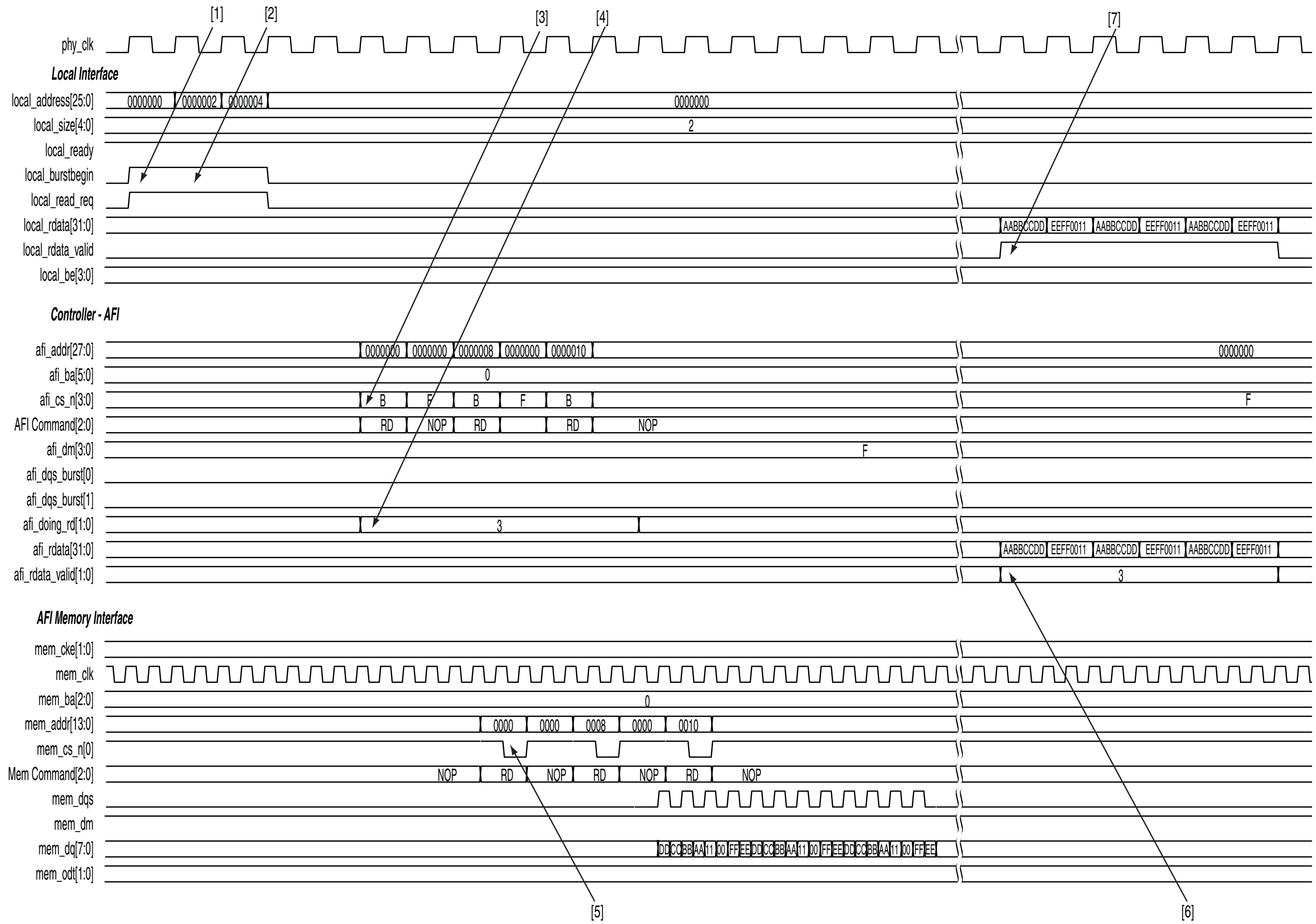
DDR and DDR2 High-Performance Controllers II

This section discusses the following timing diagrams for the DDR and DDR2 HPC II:

- "Half-Rate Read"
- "Half-Rate Write"
- "Full-Rate Read"
- "Full-Rate Write"

Half-Rate Read

Figure 17–1. Half-Rate Read Operation for HPC II



The following sequence corresponds with the numbered items in Figure 17–1:

1. The user logic requests the first read by asserting the `local_read_req` signal, and the size and address for this read. In this example, the request is a burst of length of 2 to the local address `0x000000`. This local address is mapped to the following memory address in half-rate mode:

   `mem_row_address = 0x000000`

   `mem_col_address = 0x0000`

   `mem_bank_address = 0x00`

2. The user logic initiates a second read to a different memory column within the same row. The request for the second write is a burst length of 2. In this example, the user logic continues to accept commands until the command queue is full. When the command queue is full, the controller deasserts the `local_ready` signal. The starting local address 0x000002 is mapped to the following memory address in half-rate mode:

   `mem_row_address = 0x0000`

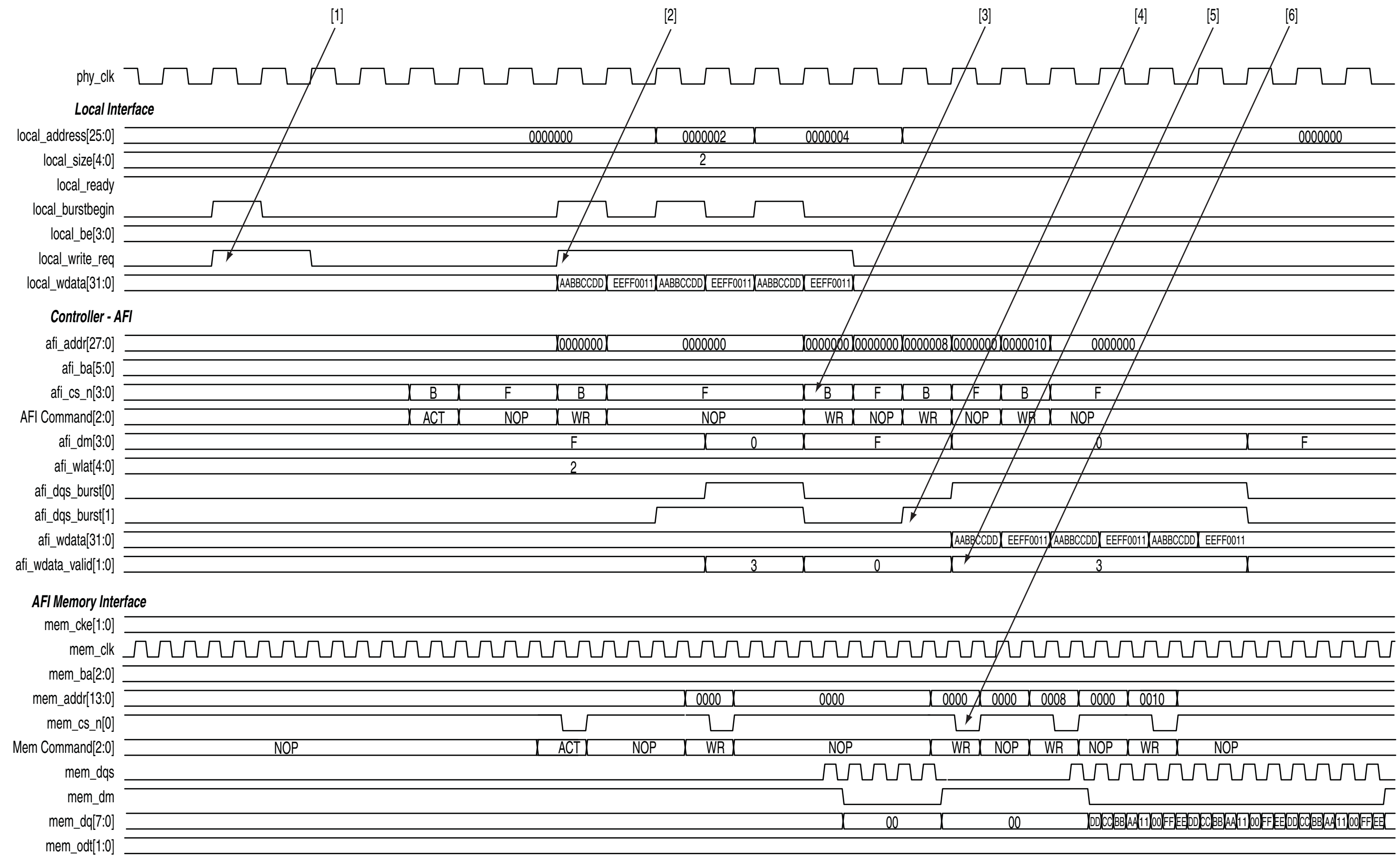
   `mem_col_address = 0x0002<<2 = 0x0008`

   `mem_bank_address = 0x00`

3. The controller issues the first read memory command and address signals to the ALTMEMPHY megafunction for it to send to the memory device.
4. The controller asserts the `afi_doing_rd` signal to indicate to the ALTMEMPHY megafunction the number of clock cycles of read data it must expect for the first read. The ALTMEMPHY megafunction uses the `afi_doing_rd` signal to enable its capture registers for the expected duration of memory burst.
5. The ALTMEMPHY megafunction issues the first read command to the memory and captures the read data from the memory.
6. The ALTMEMPHY megafunction returns the first data read to the controller after resynchronizing the data to the `phy_clk` domain, by asserting the `afi_rdata_valid` signal when there is valid read data on the `afi_rdata` bus.
7. The controller returns the first read data to the user by asserting the `local_rdata_valid` signal when there is valid read data on the `local_rdata` bus. If the ECC logic is disabled, there is no delay between the `afi_rdata` and the `local_rdata` buses. If there is ECC logic in the controller, there is one or three clock cycles of delay between the `afi_rdata` and `local_rdata` buses.

Half-Rate Write

Figure 17–2. Half-Rate Write Operation for HPC II

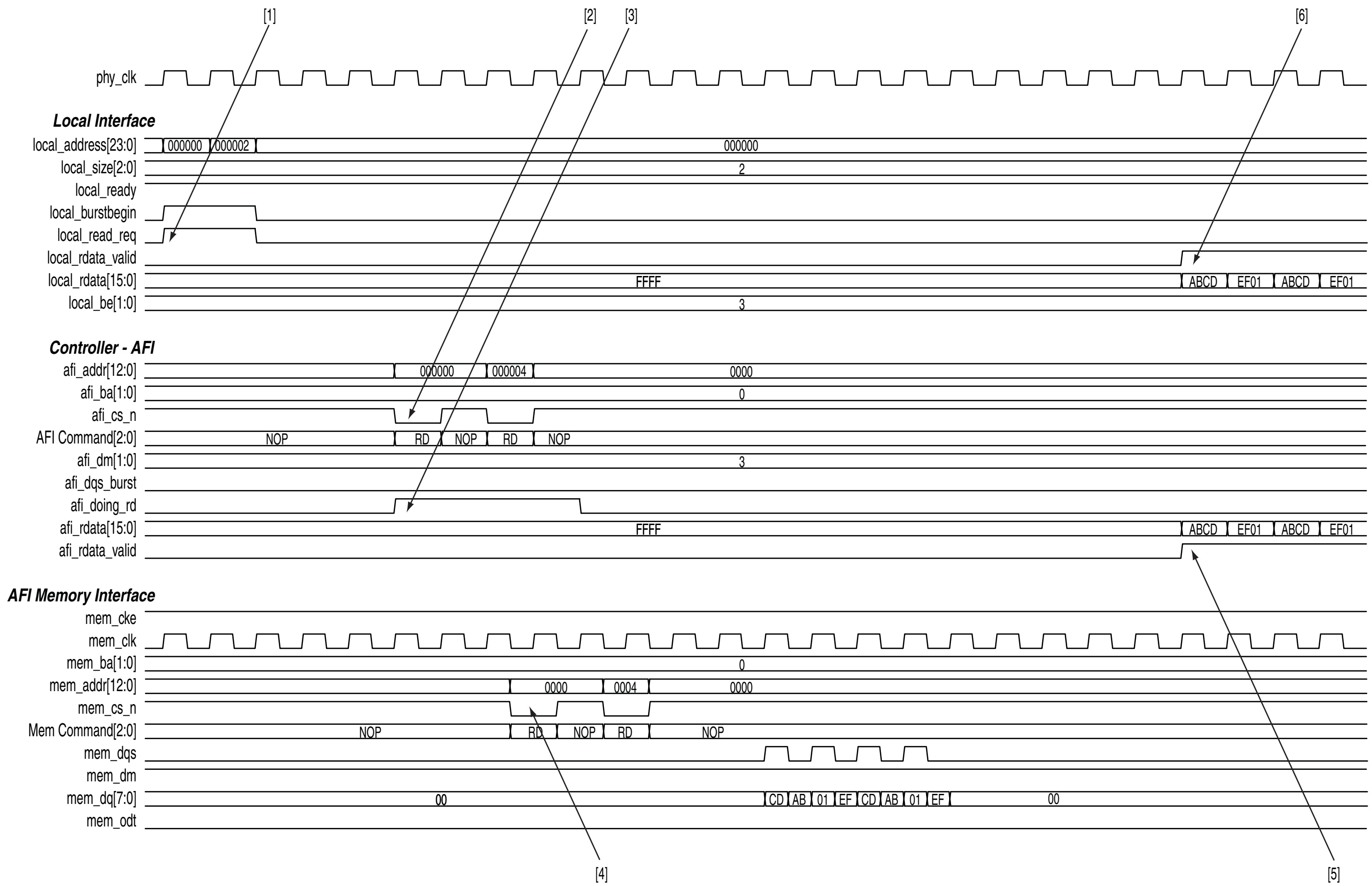


The following sequence corresponds with the numbered items in Figure 17–2:

1. The user logic asserts the first write request to row 0 so that row 0 is open before the next transaction.
2. The user logic asserts a second `local_write_req` signal with size of 2 and address of 0 (col = 0, row = 0, bank = 0, chip = 0). The `local_ready` signal is asserted along with the `local_write_req` signal, which indicates that the controller has accepted this request, and the user logic can request another read or write in the following clock cycle. If the `local_ready` signal was not asserted, the user logic must keep the write request, size, and address signals asserted until the `local_ready` signal is registered high.
3. The controller issues the necessary memory command and address signals to the ALTMEMPHY megafunction for it to send to the memory device.
4. The controller asserts the `afi_wdata_valid` signal to indicate to the ALTMEMPHY megafunction that valid write data and write data masks are present on the inputs to the ALTMEMPHY megafunction.
5. The controller asserts the `afi_dqs_burst` signals to control the timing of the DQS signal that the ALTMEMPHY megafunction issues to the memory.
6. The ALTMEMPHY megafunction issues the write command, and sends the write data and write DQS to the memory.

Full-Rate Read

Figure 17–3. Full-Rate Read Operation for HPC II



The following sequence corresponds with the numbered items in Figure 17–3:

1. The user logic requests the first read by asserting `local_read_req` signal, and the size and address for this read. In this example, the request is a burst length of 2 to a local address `0x000000`. This local address is mapped to the following memory address in full-rate mode:

   `mem_row_address = 0x0000`

   `mem_col_address = 0x0000<<2 = 0x0000`

   `mem_bank_address = 0x00`

2. The controller issues the first read memory command and address signals to the ALTMEMPHY megafunction for it to send to the memory device.

3. The controller asserts the `afi_doing_rd` signal to indicate to the ALTMEMPHY megafunction the number of clock cycles of read data it must expect for the first read. The ALTMEMPHY megafunction uses the `afi_doing_rd` signal to enable its capture registers for the expected duration of memory burst.

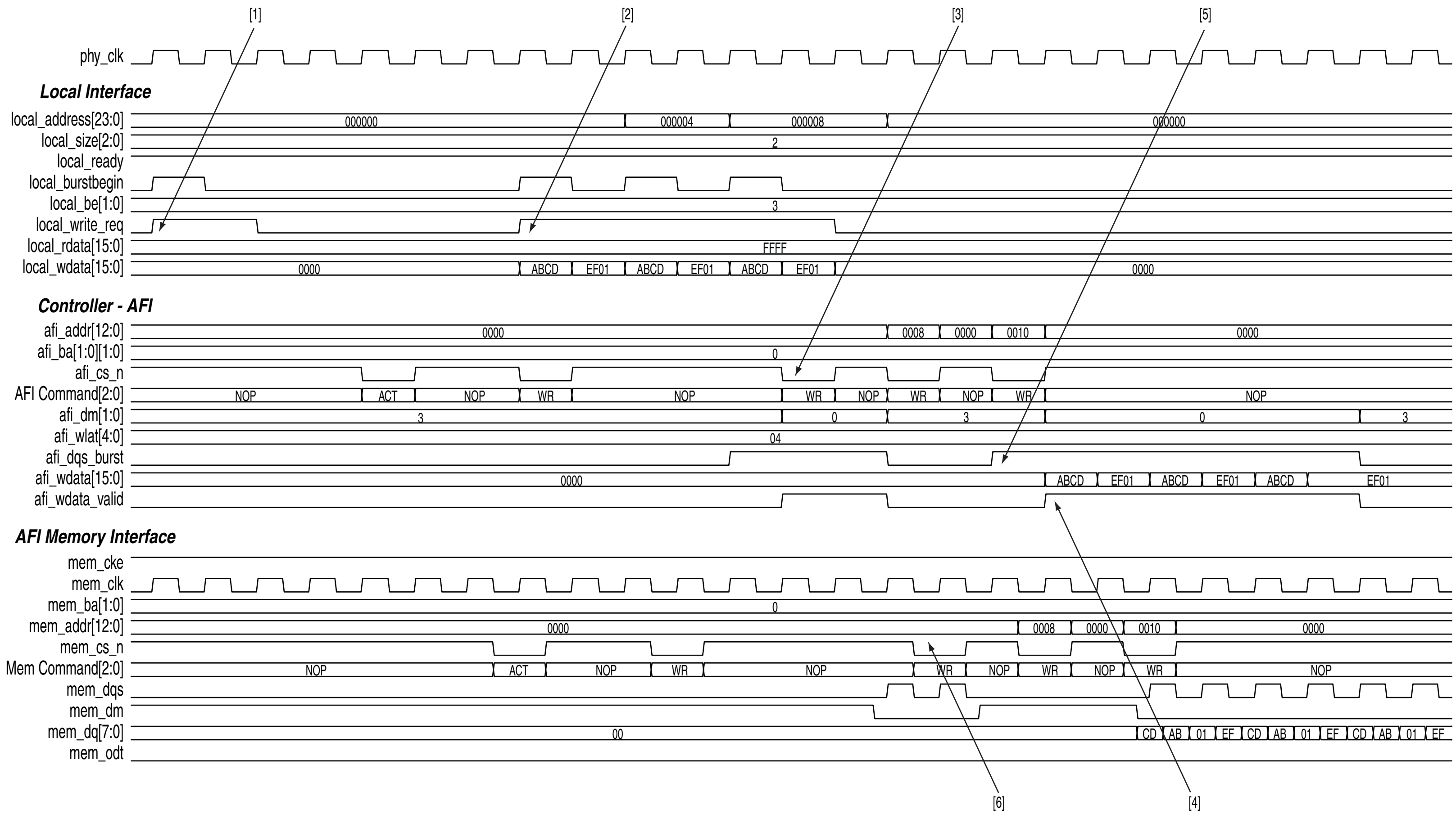
4. The ALTMEMPHY megafunction issues the first read command to the memory and captures the read data from the memory.

5. The ALTMEMPHY megafunction returns the first data read to the controller after resynchronizing the data to the `phy_clk` domain, by asserting the `afi_rdata_valid` signal when there is valid read data on the `afi_rdata` bus.

6. The controller returns the first read data to the user by asserting the `local_rdata_valid` signal when there is valid read data on the `local_rdata` bus. If the ECC logic is disabled, there is no delay between the `afi_rdata` and the `local_rdata` buses. If there is ECC logic in the controller, there is one or three clock cycles of delay between the `afi_rdata` and `local_rdata` buses.

Full-Rate Write

Figure 17–4. Full-Rate Write Operation for HPC II



The following sequence corresponds with the numbered items in Figure 17–4:

1. The user logic asserts the first write request to row 0 so that row 0 is open before the next transaction.
2. The user logic asserts a second `local_write_req` signal with a size of 2 and address of 0 (col = 0, row = 0, bank = 0, chip = 0). The `local_ready` signal is asserted along with the `local_write_req` signal, which indicates that the controller has accepted this request, and the user logic can request another read or write in the following clock cycle. If the `local_ready` signal was not asserted, the user logic must keep the write request, size, and address signals asserted until the `local_ready` signal is registered high.
3. The controller issues the necessary memory command and address signals to the ALTMEMPHY megafunction for it to send to the memory device.
4. The controller asserts the `afi_wdata_valid` signal to indicate to the ALTMEMPHY megafunction that valid write data and write data masks are present on the inputs to the ALTMEMPHY megafunction.
5. The controller asserts the `afi_dqs_burst` signals to control the timing of the DQS signal that the ALTMEMPHY megafunction issues to the memory.
6. The ALTMEMPHY megafunction issues the write command, and sends the write data and write DQS to the memory.

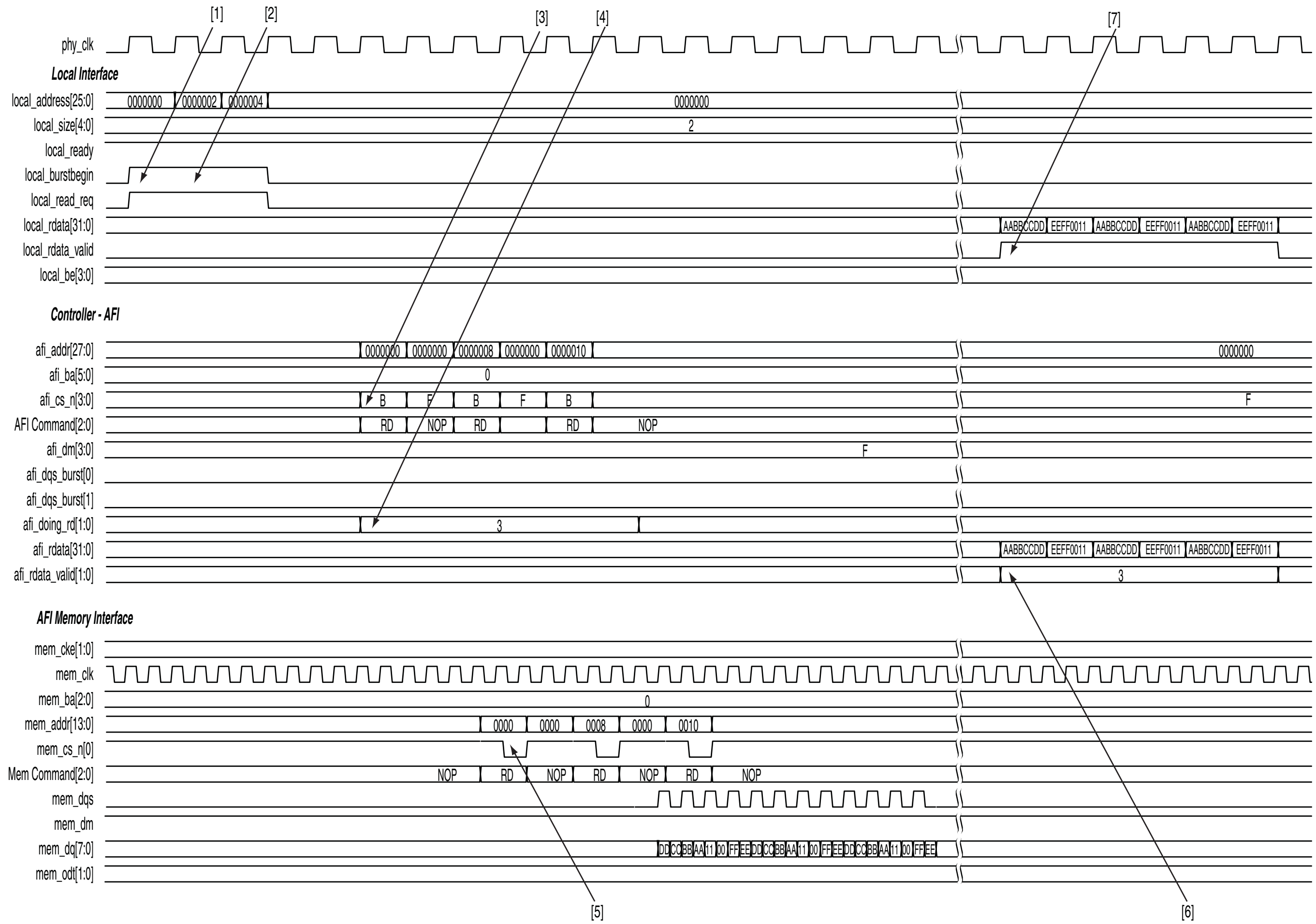
DDR3 High-Performance Controller II

This section discusses the following timing diagrams for the DDR3 HPC II:

- “Half-Rate Read”
- “Half-Rate Write”
- “Half-Rate Read (Non Burst-Aligned Address)”
- “Half-Rate Write (Non Burst-Aligned Address)”
- “Half-Rate Read With Gaps”
- “Full-Rate Read”
- “Half-Rate Write Operation (Merging Writes)”
- “Write-Read-Write-Read Operation”

Half-Rate Read (Burst-Aligned Address)

Figure 17–5. Half-Rate Read Operation for HPC II—Burst-Aligned Address



The following sequence corresponds with the numbered items in Figure 17–1:

1. The user logic requests the first read by asserting the `local_read_req` signal, and the size and address for this read. In this example, the request is a burst of length of 2 to the local address `0x000000`. This local address is mapped to the following memory address in half-rate mode:

   `mem_row_address = 0x000000`

   `mem_col_address = 0x0000`

   `mem_bank_address = 0x00`

2. The user logic initiates a second read to a different memory column within the same row. The request for the second read is a burst length of 2. In this example, the user logic continues to accept commands until the command queue is full. When the command queue is full, the controller deasserts the `local_ready` signal. The starting local address 0x000002 is mapped to the following memory address in half-rate mode:

   `mem_row_address = 0x0000`

   `mem_col_address = 0x0002<<2 = 0x0008`

   `mem_bank_address = 0x00`

3. The controller issues the first read memory command and address signals to the ALTMEMPHY megafunction for it to send to the memory device.

4. The controller asserts the `afi_doing_rd` signal to indicate to the ALTMEMPHY megafunction the number of clock cycles of read data it must expect for the first read. The ALTMEMPHY megafunction uses the `afi_doing_rd` signal to enable its capture registers for the expected duration of memory burst.

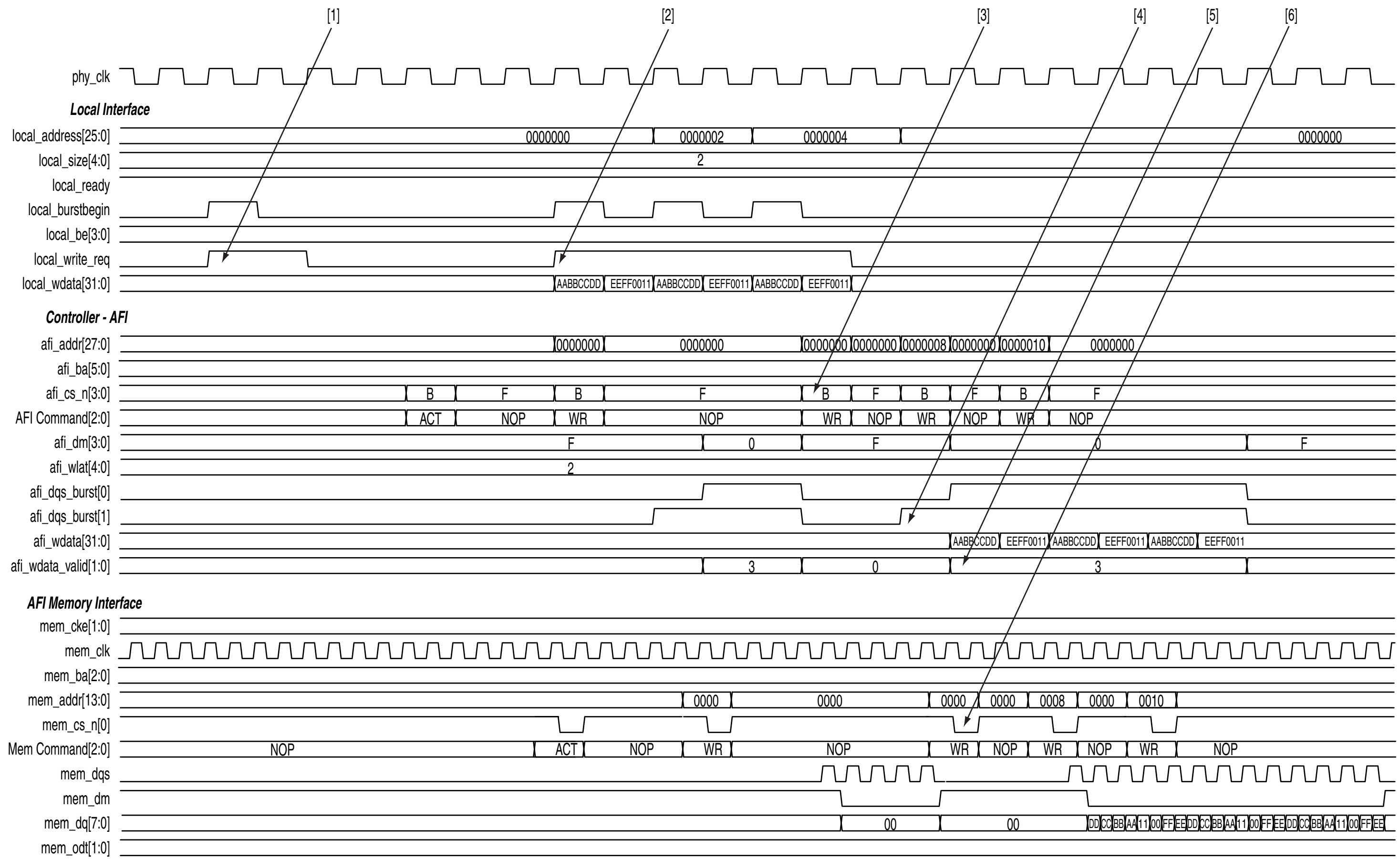
5. The ALTMEMPHY megafunction issues the first read command to the memory and captures the read data from the memory.

6. The ALTMEMPHY megafunction returns the first data read to the controller after resynchronizing the data to the `phy_clk` domain, by asserting the `afi_rdata_valid` signal when there is valid read data on the `afi_rdata` bus.

7. The controller returns the first read data to the user by asserting the `local_rdata_valid` signal when there is valid read data on the `local_rdata` bus. If the ECC logic is disabled, there is no delay between the `afi_rdata` and the `local_rdata` buses. If there is ECC logic in the controller, there is one or three clock cycles of delay between the `afi_rdata` and `local_rdata` buses.

Half-Rate Write (Burst-Aligned Address)

Figure 17–6. Half-Rate Write Operation for HPC II—Burst-Aligned Address

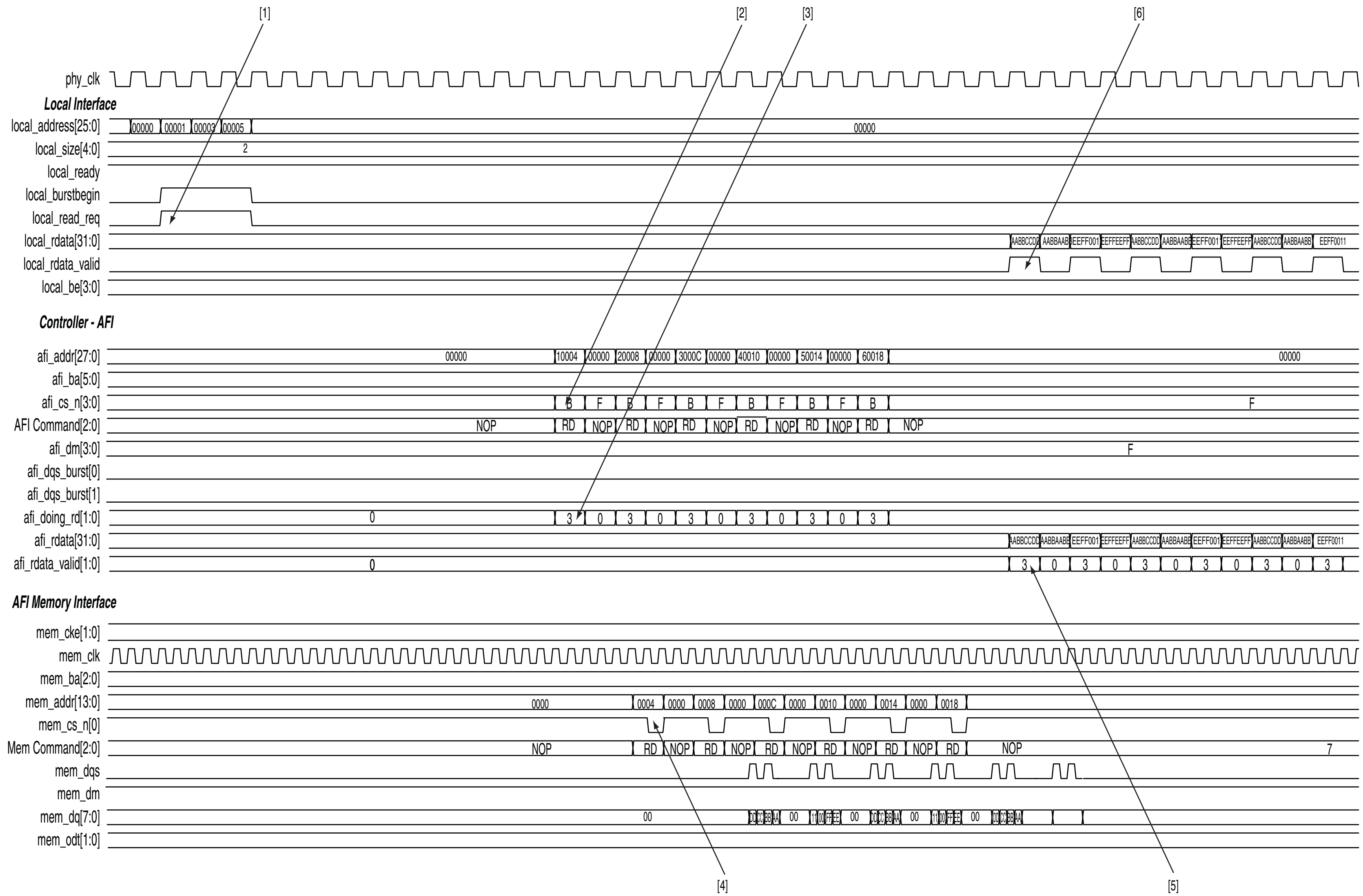


The following sequence corresponds with the numbered items in Figure 17–2:

1. The user logic asserts the first write request to row 0 so that row 0 is open before the next transaction.
2. The user logic asserts a second `local_write_req` signal with size of 2 and address of 0 (col = 0, row = 0, bank = 0, chip = 0). The `local_ready` signal is asserted along with the `local_write_req` signal, which indicates that the controller has accepted this request, and the user logic can request another read or write in the following clock cycle. If the `local_ready` signal was not asserted, the user logic must keep the write request, size, and address signals asserted until the `local_ready` signal is registered high.
3. The controller issues the necessary memory command and address signals to the ALTMEMPHY megafunction for it to send to the memory device.
4. The controller asserts the `afi_wdata_valid` signal to indicate to the ALTMEMPHY megafunction that valid write data and write data masks are present on the inputs to the ALTMEMPHY megafunction.
5. The controller asserts the `afi_dqs_burst` signals to control the timing of the DQS signal that the ALTMEMPHY megafunction issues to the memory.
6. The ALTMEMPHY megafunction issues the write command, and sends the write data and write DQS to the memory.

Half-Rate Read (Non Burst-Aligned Address)

Figure 17–7. Half-Rate Read Operation for HPC II—Non Burst-Aligned Address



The following sequence corresponds with the numbered items in Figure 17–7:

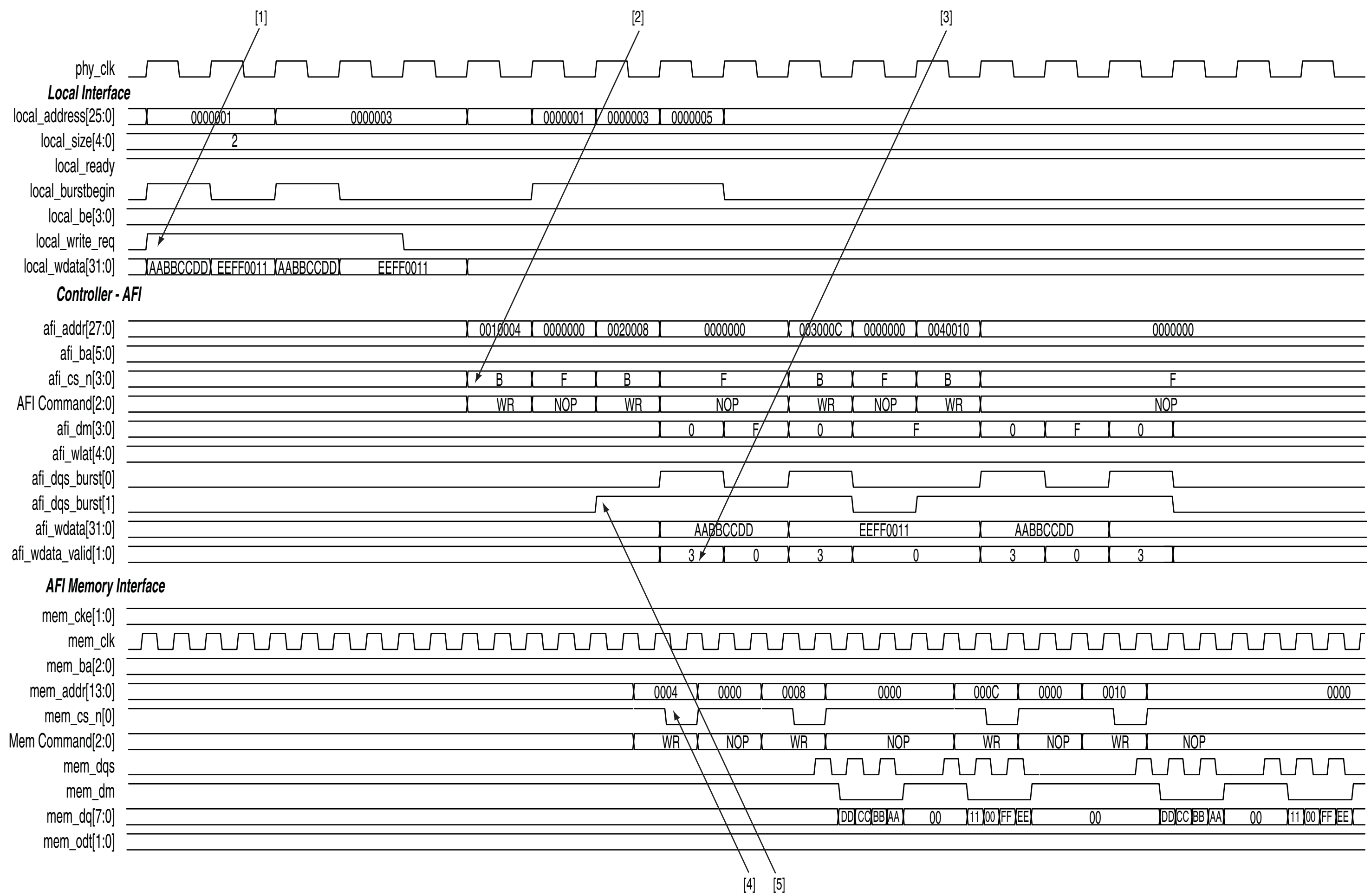
1. The user logic requests the first read by asserting the `local_read_req` signal, and the size and address for this read. In this example, the request is a burst of length of 2 to the local address `0x000001`. This local address is mapped to the following memory address in half-rate mode:

   ```
   mem_row_address = 0x0000

   mem_col_address = 0x0001<<2 = 0x0004

   mem_bank_address = 0x00
   ```

2. The controller issues the first read memory command and address signals to the ALTMEMPHY megafunction for it to send to the memory device.
3. The controller asserts the `afi_doing_rd` signal to indicate to the ALTMEMPHY megafunction the number of clock cycles of read data it must expect for the first read. The ALTMEMPHY megafunction uses the `afi_doing_rd` signal to enable its capture registers for the expected duration of memory burst.
4. The ALTMEMPHY megafunction issues the first read command to the memory and captures the read data from the memory.
5. The ALTMEMPHY megafunction returns the first data read to the controller after resynchronizing the data to the `phy_clk` domain, by asserting the `afi_rdata_valid` signal when there is valid read data on the `afi_rdata` bus.
6. The controller returns the first read data to the user by asserting the `local_rdata_valid` signal when there is valid read data on the `local_rdata` bus. If the ECC logic is disabled, there is no delay between the `afi_rdata` and the `local_rdata` buses. If there is ECC logic in the controller, there is one or three clock cycles of delay between the `afi_rdata` and `local_rdata` buses.

Half-Rate Write (Non Burst-Aligned Address)

Figure 17–8. Half-Rate Write Operation for HPC II—Non Burst-Aligned Address



The following sequence corresponds with the numbered items in Figure 17–8:

1. The user logic asserts the first `local_write_req` signal with a size of 2 and an address of `0×000001`. The `local_ready` signal is asserted along with the `local_write_req` signal, which indicates that the controller has accepted this request, and the user logic can request another read or write in the following clock cycle. If the `local_ready` signal was not asserted, the user logic must keep the write request, size, and address signals asserted until the `local_ready` signal is registered high. The local address 0x000001 is mapped to the following memory address in half-rate mode:

   `mem_row_address = 0×0000`

   `mem_col_address = 0×000001<<2 = 0×000004`

   `mem_bank_address = 0×00`

2. The user logic asserts the second `local_write_req` signal with a size of 2 and an address of `0×000003`. The local address `0×000003` is mapped to the following memory address in half-rate mode:

   `mem_row_address = 0×0000`

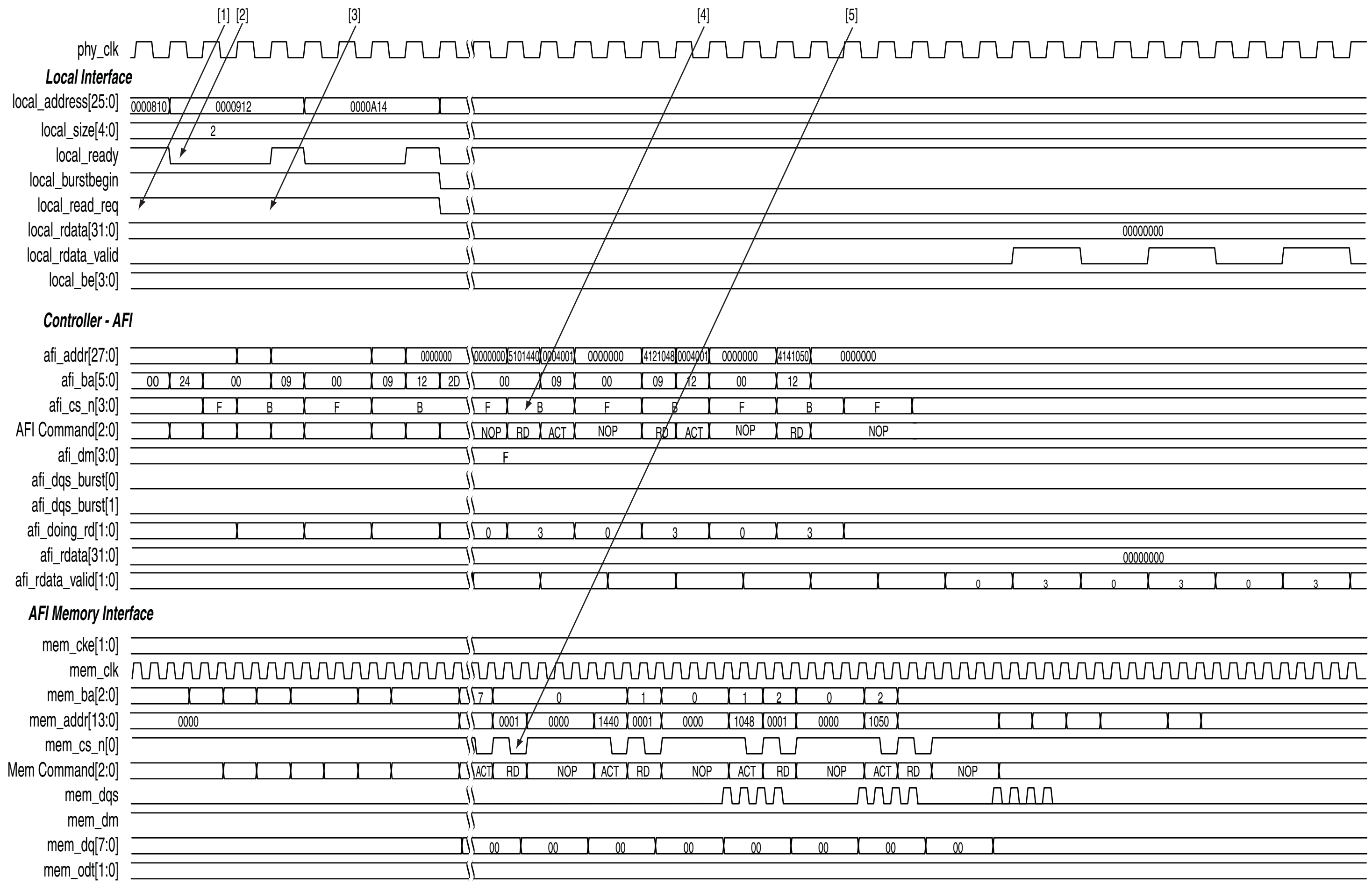
   `mem_col_address = 0×000003<<2 = 0×00000C`

   `mem_bank_address = 0×00`

3. The controller issues the necessary memory command and address signals to the ALTMEMPHY megafunction for it to send to the memory device.
4. The controller asserts the `afi_wdata_valid` signal to indicate to the ALTMEMPHY megafunction that valid write data and write data masks are present on the inputs to the ALTMEMPHY megafunction.
5. The controller asserts the `afi_dqs_burst` signals to control the timing of the DQS signal that the ALTMEMPHY megafunction issues to the memory.
6. The ALTMEMPHY megafunction issues the write command, and sends the write data and write DQS to the memory.
7. The controller generates another write because the first write is to a non-aligned memory address, `0×0004`. The controller performs the second write burst at the memory address of `0×0008`.

Half-Rate Read With Gaps

Figure 17–9. Half-Rate Read Operation for HPC II—With Gaps



The following sequence corresponds with the numbered items in Figure 17–9:

1. The user logic requests the first read by asserting the `local_read_req` signal, and the size and address for this read. In this example, the request is a burst of length of 2 to the local address `0×0000810`. This local address is mapped to the following memory address in half-rate mode:

   `mem_row_address = 0×0001`

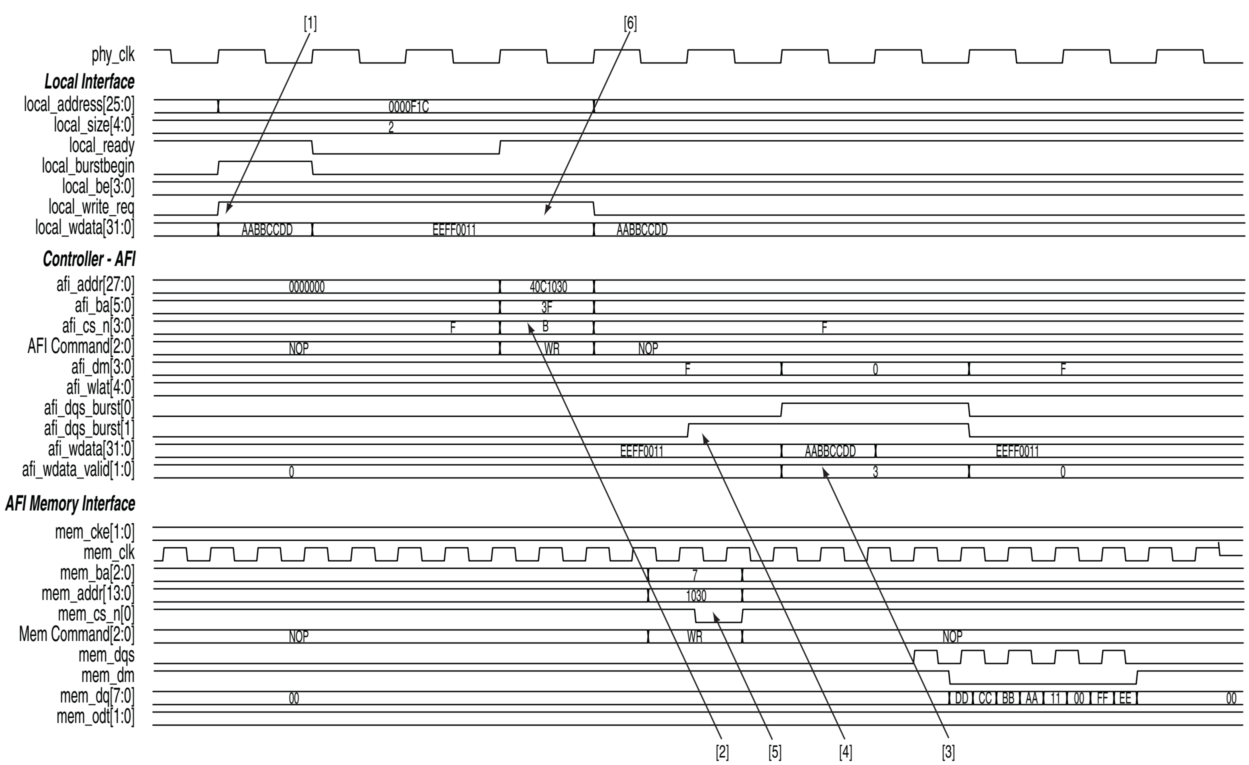
   `mem_col_address = 0×0010<<2 = 0×0040`

   `mem_bank_address = 0×00`

2. When the command queue is full, the controller deasserts the `local_ready` signal to indicate that the controller has not accepted the command. The user logic must keep the read request, size, and address signal until the `local_ready` signal is asserted again.
3. The user logic asserts a second `local_read_req` signal with a size of 2 and address of `0×0000912`.
4. The controller issues the first read memory command and address signals to the ALTMEMPHY megafunction for it to send to the memory device.
5. The ALTMEMPHY megafunction issues the read command to the memory and captures the read data from the memory.

Half-Rate Write With Gaps

Figure 17–10. Half-Rate Write Operation for HPC II—With Gaps

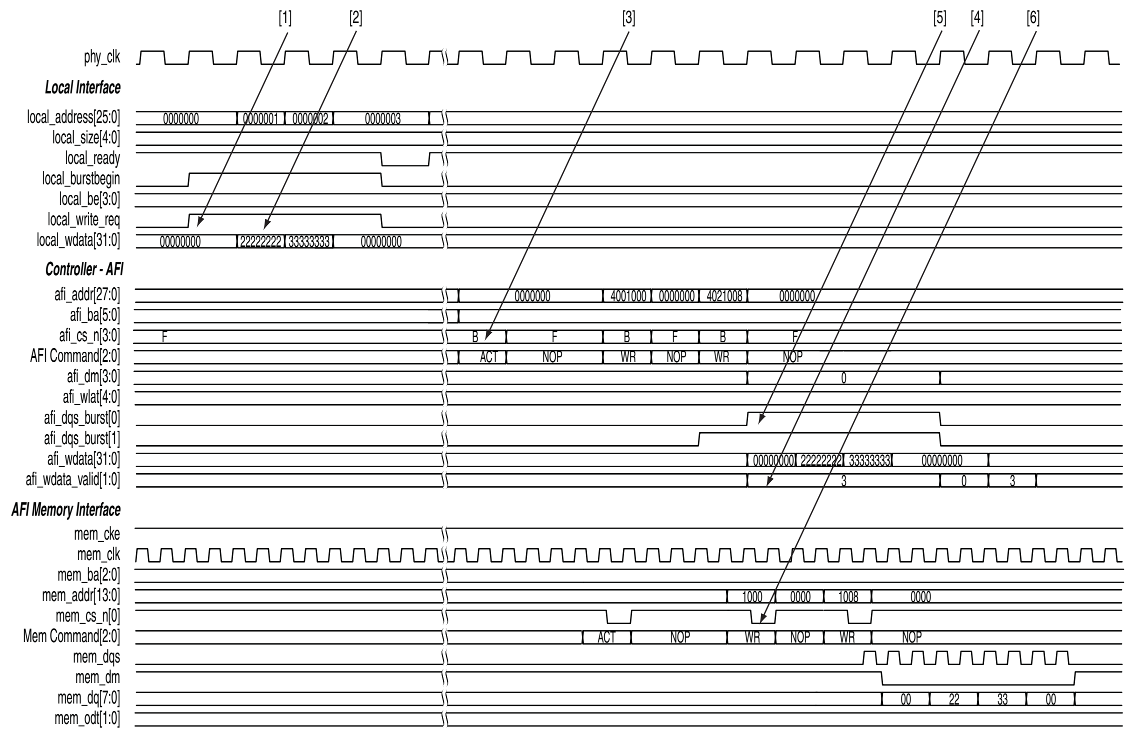


The following sequence corresponds with the numbered items in Figure 17–10:

1. The user logic asserts a `local_write_req` signal with a size of 2 and an address of `0×0000F1C`.
2. The controller issues the necessary memory command and address signals to the ALTMEMPHY megafunction for it to send to the memory device.
3. The controller asserts the `afi_wdata_valid` signal to indicate to the ALTMEMPHY megafunction that valid write data and write data masks are present on the inputs to the ALTMEMPHY megafunction.
4. The controller asserts the `afi_dqs_burst` signals to control the timing of the DQS signal that the ALTMEMPHY megafunction issues to the memory.
5. The ALTMEMPHY megafunction issues the write command, and sends the write data and write DQS to the memory.
6. For transactions with a local size of two, the `local_write_req` and `local_ready` signals must be high for two clock cycles so that all the write data can be transferred to the controller.

Half-Rate Write Operation (Merging Writes)

Figure 17–11. Write Operation for HPC II—Merging Writes



The following sequence corresponds with the numbered items in Figure 17–11:

1. The user logic asserts the first `local_write_req` signal with a size of 1 and an address of `0×000000`. The `local_ready` signal is asserted along with the `local_write_req` signal, which indicates that the controller has accepted this request, and the user logic can request another read or write in the following clock cycle. If the `local_ready` signal was not asserted, the user logic must keep the write request, size, and address signals asserted until the `local_ready` signal is registered high. The local address `0x000000` is mapped to the following memory address in half-rate mode:

   `mem_row_address = 0×0000`

   `mem_col_address = 0×0000<<2 = 0×0000`

   `mem_bank_address = 0×00`

2. The user logic asserts a second `local_write_req` signal with a size of 1 and address of 1. The `local_ready` signal is asserted along with the `local_write_req` signal, which indicates that the controller has accepted this request. Since the second write request is to a sequential address (same row, same bank, and column increment by 1), this write and the first write can be merged at the memory transaction.

3. The controller issues the necessary memory command and address signals to the ALTMEMPHY megafunction for it to send to the memory device.

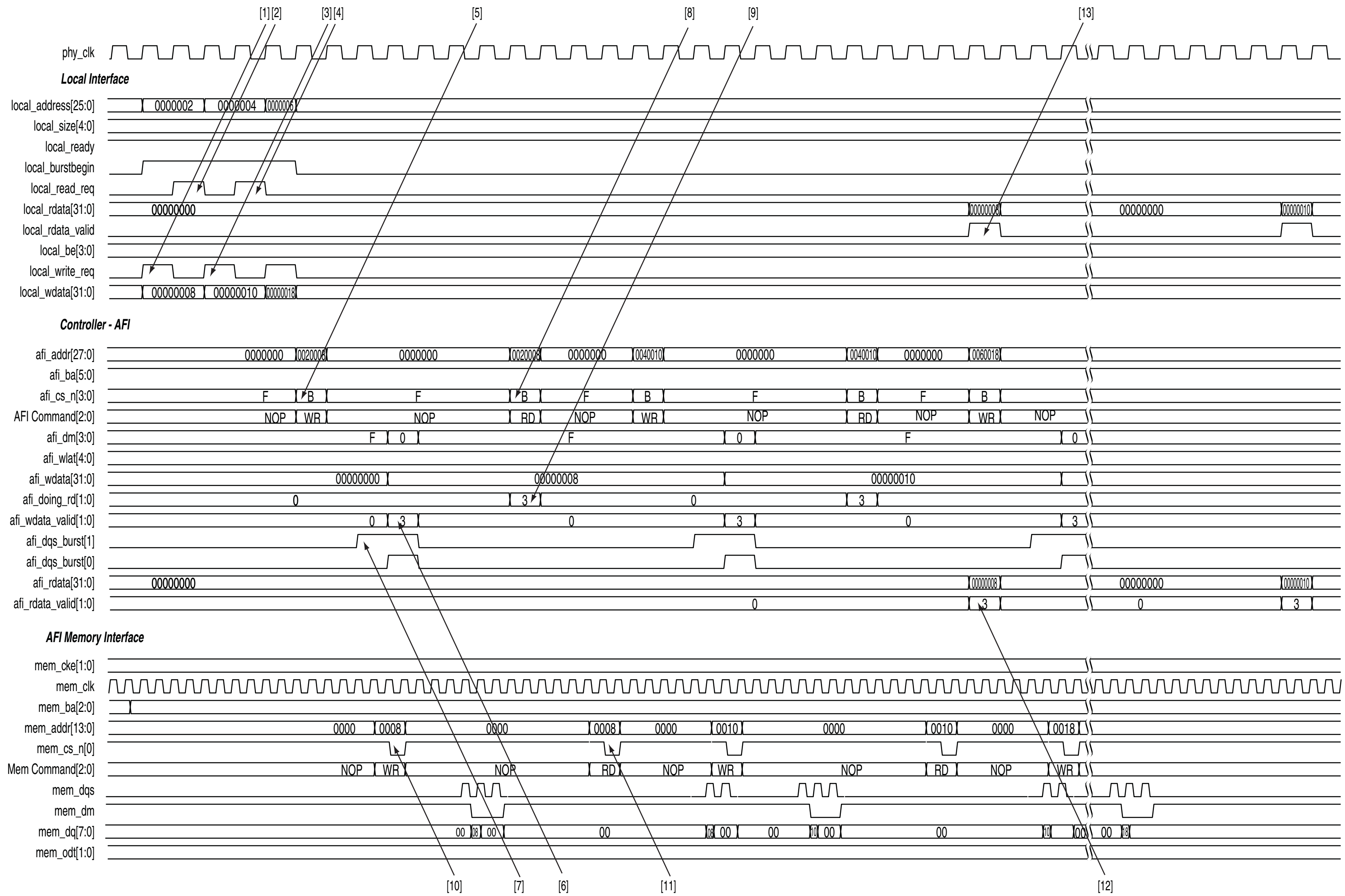
4. The controller asserts the `afi_wdata_valid` signal to indicate to the ALTMEMPHY megafunction that valid write data and write data masks are present on the inputs to the ALTMEMPHY megafunction.

5. The controller asserts the `afi_dqs_burst` signals to control the timing of the DQS signal that the ALTMEMPHY megafunction issues to the memory.

6. The ALTMEMPHY megafunction issues the write command, and sends the write data and write DQS to the memory.

Write-Read-Write-Read Operation

Figure 17–12. Write-Read Sequential Operation for HPC II



The following sequence corresponds with the numbered items in Figure 17–12:

1. The user logic requests the first write by asserting the `local_write_req` signal, and the size and address for this write. In this example, the request is a burst length of 1 to a local address `0x000002`. This local address is mapped to the following memory address in half-rate mode:

   `mem_row_address = 0x0000`

   `mem_col_address = 0x0002<<2 = 0x0008`

   `mem_bank_address = 0x00`

2. The user logic initiates the first read to the same address as the first write. The request for the read is a burst length of 1. The controller continues to accept commands until the command queue is full. When the command queue is full, the controller deasserts the `local_ready` signal. The starting local address 0x000002 is mapped to the following memory address in half-rate mode:

   `mem_row_address = 0x0000`

   `mem_col_address = 0x0002<<2 = 0x0008`

   `mem_bank_address = 0x00`

3. The user logic asserts a second `local_write_req` signal with a size of 1 and address of `0x000004`.
4. The user logic asserts a second `local_read_req` signal with a size of 1 and address of `0x000004`.
5. The controller issues the necessary memory command and address signals to the ALTMEMPHY megafunction for it to send to the memory device.
6. The controller asserts the `afi_wdata_valid` signal to indicate to the ALTMEMPHY megafunction that valid write data and write data masks are present on the inputs to the ALTMEMPHY megafunction.
7. The controller asserts the `afi_dqs_burst` signals to control the timing of the DQS signals that the ALTMEMPHY megafunction issues to the memory.
8. The controller issues the first read memory command and address signals to the ALTMEMPHY megafunction for it to send to the memory device.
9. The controller asserts the `afi_doing_rd` signal to indicate to the ALTMEMPHY megafunction the number of clock cycles of read data it must expect for the first read. The ALTMEMPHY megafunction uses the `afi_doing_rd` signal to enable its capture registers for the expected duration of memory burst.
10. The ALTMEMPHY megafunction issues the write command, and sends the write data and write DQS to the memory.
11. The ALTMEMPHY megafunction issues the first read command to the memory and captures the read data from the memory.
12. The ALTMEMPHY megafunction returns the first data read to the controller after resynchronizing the data to the `phy_clk` domain, by asserting the `afi_rdata_valid` signal when there is valid read data on the `afi_rdata` bus.

13. The controller returns the first read data to the user by asserting the `local_rdata_valid` signal when there is valid read data on the `local_rdata` bus. If the ECC logic is disabled, there is no delay between the `afi_rdata` and the `local_rdata` buses. If there is ECC logic in the controller, there is one or three clock cycles of delay between the `afi_rdata` and `local_rdata` buses.

Document Revision History

Table 17–1 lists the revision history for this document.

Table 17–1. Document Revision History

Date	Version	Changes
November 2012	1.3	Changed chapter number from 15 to 17.
June 2012	1.2	Added Feedback icon.
November 2011	1.1	Consolidated timing diagrams from 11.0 version **DDR and DDR2 SDRAM Controller with ALTMEMPHY IP User Guide** and **DDR3 SDRAM Controller with ALTMEMPHY IP User Guide**.