Striving to achieve an integrated user experience, today’s devices are getting crammed with loads of features which operate on voluminous data traffic over various interfaces. For efficient processing of these data, faster memories offering high bandwidth are the need of the hour. In spite of availability of many different kinds of memories, Double Data Rate (DDR) memories maintain their dominant position when it comes to offering large amount of dynamic random access storage with high bandwidth data transfer interface. These types of memories are called Double Data Rate as they offer double the performance compared to the Single Data Rate memories by allowing two data transactions per memory clock.

A typical DDR memory is arranged in banks having multiple rows and columns along with pre-fetch buffers. For any data transaction, the memory address is split into bank address, row address and column addresses. The performance advantages of the DDR memory are mainly due to its pre-fetch architecture with burst oriented operation; where a memory access to a particular row of a bank causes the pre-fetch buffer to grab a set of adjacent datawords and subsequently burst them on IO pins on each edge of the memory clock, without requiring individual column addresses. Thus the higher the size of pre-fetch buffers, the higher is the bandwidth. Higher bandwidth is also achieved by creating modules with multiple DDR memory chips.

DDR memories require specific power up and initialization sequence prior to their operation. Before any read or write transaction, a particular row of a bank needs to be activated/opened (which essentially activates and amplifies the signals from that row) and after the end of the transaction it is pre-charged/closed if no further access to the row is needed. The DDR memories need to be periodically refreshed so that they don’t lose any of their contents.

The size of pre-fetch buffer is 2^n (two datawords per memory access) for DDR memories, 4^n (four datawords per memory access) for DDR2 memories and 8^n (eight datawords per memory access) for DDR3 memories; where n is the size of IO interface typically 4, 8 or 16. These pre-fetch schemes attribute their effectiveness to the principle of spatial locality.

With these basic understandings, the specific features and functionalities of DDR3 memories are further discussed in the following sections.

**DDR3 Memory**

DDR3 memories provide much improved performance compared to DDR2 memories due to their low power, higher clock frequency operation along with 8^n pre-fetch architecture offering significantly higher bandwidth for data transfers. Typically a DDR3 memory operates at 1.5V at 400-800MHz memory clock frequency; thus offering a data rate per pin ranging from 800-1600Mbps. DDR3 memories are available in IO interface sizes of 4, 8 and 16; supporting burst lengths of 4 and 8 datawords per memory access. The important features of DDR3 memories are compared with those of DDR2 memories in Table 1.
Besides the above improvements in feature, DDR3 memories incorporate the following additional/new specifications which differ from the DDR2 memories:

- Introduction of Fly-by routing for connecting command and address signals to memory modules to provide improved signal integrity at high speeds.
- Write leveling and Read Leveling for skew compensation due to Fly-by routing.
- Incorporation of dedicated ZQ pad and ZQ calibration sequences along with on-die calibration engine for calibrating the On-Die termination circuit and output driver.
- Dedicated RESET pin
- Enhanced low power feature
- Dynamic On-Die termination to improve signal integrity for write transactions.

The following sections describe the above specifications in greater detail. **Fly-By Topology**

**Fly-By Topology**
The higher signaling rates of DDR3 necessitated a new topology for routing the command and control signals to different memory modules. The T-topology, shown in Figure 1, which was adopted for DDR2 couldn’t support higher signaling rates and more number of memory modules due to capacitive loading. In a T-topology, the signals are routed to a central node before routing them to individual memory modules, thus limiting the potential variability of trace lengths to shorter paths. But higher signaling rates couldn’t be reliably supported over this topology due to multiple stubs and increase in capacitive load seen by the signals when increasing memory capacity.

![Fly-By Topology](Click for larger image)

**Figure 1.** Shows T-topology for connecting memory controller and DDR2 memory modules in which the Command/Address/Clock signals are routed to each memory module in a branched fashion.

The above problems are overcome by adopting Fly-by topology for DDR3, which connects the command and address signals in series with each of the memory modules along with appropriate termination at the end. The signals travelling in this topology reach different memory modules at
different time intervals and encounter the input capacitive load of the memory modules in a delayed fashion. Thus the capacitive load is reduced paving the way for higher signaling rates and scalable memory systems without compromising on the data rates. Figure 2 illustrates the Fly-by topology adopted for DDR3 memory systems.

Figure 2. Depicts Fly-by topology for connecting memory controller and DDR3 memory modules in which the memory modules share common Command/Address/Clock lines connected in series.

Write Leveling
Due to the Fly-by topology of DDR3 memories, data and strobes reach different memory modules at different times with respect to command, address and clock signals. To address this situation, DDR3 memories implement leveling techniques, which align the data strobes with clock signal at each memory module interface. Leveling is carried out for each memory module present in a system for each data byte.

Write leveling remedies the skew between data strobes and clock at the memory module boundaries for write data transactions. Before staring the write leveling, the DDR3 memory is placed in the write leveling mode by writing appropriate mode register. After placing the memory in write leveling mode, clock and data strobes are given to the memory module. The memory module samples the clock signal at its boundary with the observed data strobes and the feeds back the sampled value (0/1) on data lines to the driving entity so that it can adjust the delay in the data strobes for next iteration. This process is repeated till a 0 to 1 transition in the feedback value is observed, which indicates the alignment of data strobes with respect to the clock signal at the memory module boundary. The write leveling process is shown in Figure 3 as a waveform diagram.
Figure 3. Illustrates the Write Leveling process for adjusting skews at each memory module between data and command/address/clock signals by progressively altering the delay on data strobe line till a 0 to 1 transition in memory clock is sampled at the targeted memory module using the delayed data strobe.

**Figure 3 Legend**

- **D1**: Delay in the clock observed at the targeted memory module with respect to the clock at the controller end
- **D2**: Delay added to the data strobe at the controller side to observe a 0 to 1 transition in the clock at the targeted memory module
- **D3**: Delay in the data strobe signal observed at the targeted memory module with respect to the data strobe signal at the controller end
- **D4**: Delay between the sampled value of clock at the targeted memory module for previous data strobe adjustment and driving data strobe with new adjustment value

**Read Leveling**

Read leveling addresses the skew between data and strobes for read data transactions. To support this feature, DDR3 memories incorporate a Multi Purpose Register (MPR) which contains a pre-defined data pattern which, when selected, is output on the data lines, instead of normal data from the memory array. Before starting the read leveling sequence, the MPR data is selected as output by programming appropriate mode register. Thereafter read leveling is initiated by giving READ command to the memory module and trying to capture the pre-defined data by optimally adjusting the internal delays on data strobes. This process is repeated till the internal delays on the data strobes are adjusted to create a proper window for best capture the pre-defined data pattern. The read leveling process is shown as a waveform diagram in Figure 5 and the provision for selecting the Multi Purpose Register (MPR) for read leveling process is depicted by Figure 4.
Both Write and Read leveling are relevant for DDR3 memories only. DDR2 memories don’t have any such provisions.

**ZQ Calibration**
To improve signal integrity and boost the output signals DDR memories come with termination resistances and output drivers. Periodic calibration of these termination resistances and output drivers are necessary for maintaining signal integrity across temperature and voltage variations. While uncalibrated termination resistances directly affect the signal quality, improperly adjusted output drivers shift the valid signal transitions from reference level; thus causing skew between data and strobe signals. As shown in Figure 6, this skew reduces the effective valid data window and decreases the reliability of data transfers.
The output drivers of DDR2 memories are generally present off-chip. These off-chip drivers are optionally calibrated only once during initialization. This calibration sequence, known as Off-Chip Driver calibration, only calibrates the output driver present off-chip. On-die termination calibration doesn’t happen for DDR2 memories.

In order to maintain high signal integrity, DDR3 memories incorporate on-die terminations (ODT) and on-chip output drivers. A dedicated pad, known as ZQ pad, is present in the DDR3 memories which facilitates the calibration process through a 240 ohm ± 1% tolerance external resistor between the ZQ pin and ground acting as reference. The calibration sequence is initiated by the on-die calibration engine when the memory module receives a ZQ calibration command. Initial ZQ calibration is done during initialization and short ZQ calibrations are done periodically to compensate variations in operating temperature and voltage drifts.

**Dynamic On-Die Termination**

DDR3 memories offer a new feature where the on-die termination resistance can be changed without mode register programming in order to improve the signal integrity on the data bus. When this feature is enabled a different value of termination resistance is applied for write data transactions to the memory. Figure 7 represents an abstract arrangement implemented in DDR3 memories which dynamically switches the termination resistance, when enabled, for write transactions; thus eliminating the need of issuing mode register programming command in such cases.
Dedicated RESET Pin

Dedicated RESET Pin
DDDR3 memories have a dedicated RESET pin to asynchronously reset the internal states of the memory in event of error conditions.

Low Power Modes
Like DDR2 memories, DDR3 memories provide power down and self refresh modes to conserve power when not in use. In self refresh mode, DDR3 memory retains data without external clocking while the rest of the system is powered down. When the memory is not accessed for a longer duration of time, it can be put in power down mode, by making the CKE signal LOW, where it doesn’t retain any of the data. When the power down happens while all the memory banks are pre-charged, it is called pre-charge power down and in case if any of the memory banks are active during power down, then it is called active power down. The memory can be taken out of power down mode by driving the CKE signal HIGH. The low power mode transitions are controlled by the memory controller, thus providing the flexibility of putting the memory modules in low power states and exiting from them, as per requirement. Typically DDR3 memories enter the desired low power mode one memory clock cycle after receiving the appropriate command and exit from it when necessary conditions are met. As per the JEDEC specification, the total power down entry and exit timings should be minimum 7.5ns for DDR3-800 type memories. For more details the DDR3 JEDEC specification can be referred.

Guidelines for an efficient DDR3 Memory Controller
In order to manage various DDR3 memory features and to provide an abstracted, bandwidth efficient and automated way to initialize and use the memory, an efficient DDR3 memory controller
is required. The memory controller should not only automatically initialize of the memory based on programmed controller parameters after power-on, but also should include high bandwidth interfaces with queuing, prioritization, arbitration and re-ordering capabilities for efficient, decoupled access to the memory in case of multiple simultaneous memory accesses.

A typical DDR3 memory controller sub-system consists of the following components as shown in Figure 8:

1. High bandwidth interface(s) for catering to memory access requests
2. Register access port for controller parameter configuration for memory initialization
3. Core controller module containing queues and memory command processing engine
4. PHY interface for driving the input memory transaction to the memory PHY
5. Memory PHY driving access requests to memory as per DDR3 protocol
6. ASIC pad calibration logic to maintain proper voltage level on the memory interface

The following sections shed more light on the above controller building blocks.

**Memory Access Interface**
The DDR3 memory controller provides memory access interfaces and controllers for external systems requiring access to the memory. The memory interface should support high bandwidth and high frequency operation to efficiently utilize the DDR3 memories. Multiple memory access interfaces can be implemented to cater to multiple simultaneous access requests. Apart from the memory location address for write/read data transactions and their enables, this interface protocol should contain information about the access requesting entity and response mechanism for each data transaction received. The interface protocol needs to be burst oriented to fully utilize the burst oriented DDR3 memories. In case of multiple access interfaces, the interface protocol should have a
priority field to indicate the priority of each data transaction.

**Register Access Interface**

**Register Access Interface**
The register access interface enables the programmer to configure controller parameters for a particular DDR3 memory initialization at power up. Since this interface is not necessarily required to operate at high frequency, it can be implemented as per specific requirement. This interface can optionally have an error indication for attempts to program an invalid controller register.

**Core Controller Module**

This module of the controller is responsible for processing any data access requests on the memory access interface and after proper formatting sending them across to the memory PHY to be driven on to the memory. In order to carry out the different tasks involved, this module can be further divided into the following sub-modules:

a. Memory access interface blocks
b. Arbitration Unit
c. Queues with placement logic for holding commands and write/read data queues
d. Command processing unit

Memory access interface blocks decode the memory access requests from external systems and store them in their internal FIFOs. The data access requests can be split into write/read commands and their respective data with their priority. The write/read data can be stored in separate dedicated FIFOs and the associated commands can be stored in a command FIFO after arbitrating them according to the relative priority of read/write transactions. An error indication for the external systems can be implemented in case of any errors associated with a particular command received on the interface. The read, write and command FIFO depths can be made configurable according to the traffic on the interface.

Arbitration unit selects commands from multiple memory access interface blocks and sends then to command queue to be driven onto the memory interface after processing. A suitable arbitration scheme, like round-robin arbitration, can be selected for scanning the Memory access interface blocks for commands. The commands can then be divided into priority groups as per the priority of the originating port; after which a single high priority command can be send to the command queue. In order to avoid situations where low priority commands may not get a chance to be executed on continuous high priority command receptions, a priority group disabling feature can be incorporated which will disable a particular priority group in a time-controlled fashion. All associated priority fields can be made user programmable so that it can be tuned as per need.

In order to understand the arbitration scheme, let us consider a core controller module having two memory access interface blocks (IF0, IF1) with two priority group levels (PG0, PG1), as shown in Figure 9.
Let the packets $P_0^0$ with priority 0 ($P_0^0, 0$) and $P_0^1$ with priority 1 ($P_0^1, 1$) be received on IF0; and packets $P_1^0$ with priority 0 ($P_1^0, 0$) and $P_1^1$ with priority 1 ($P_1^1, 1$) be received on IF1. Considering round-robin arbitration between the ports, the packets will be available for arrangement into priority groups as follows.

$P_0^0, 0 \rightarrow P_1^0, 0 \rightarrow P_0^1, 1 \rightarrow P_1^1, 1$

As the priority group 0 stores the packets with priority 0 and the priority group 1 stores the packets with priority 1, the packets $P_0^0$, 0 and $P_1^0$, 0 will be stored in priority group 0 (PG0); and packets $P_0^1$, 1 and $P_1^1$, 1 will be stored in priority group 1 (PG1).

$PG0 \rightarrow P_0^0$, 0 and $P_1^0$, 0

$PG1 \rightarrow P_0^1$, 1 and $P_1^1$, 1

Assuming that the priority group 1 (PG1) has higher weight than priority group 0 (PG0) and the packets from memory access interface block 0 (IF0) have higher weight than those from memory access interface block 1 (IF1), the packets present in the priority groups will be arranged in the command queue in the following order.

$P_0^1$, 1 $\rightarrow$ $P_1^1$, 1 $\rightarrow$ $P_0^0$, 0 $\rightarrow$ $P_1^0$, 0

**Command queue**

After arbitration, the commands can be placed into a queue in which they can be re-ordered for efficient operation and collision avoidance. A re-ordering unit can be implemented which looks into potential scenarios of address/source/data collisions among the commands in the queue and arranges them according to their type and priority. Additionally the commands can be ordered for efficient controller operation, e.g., by allowing commands accessing different memory banks between commands operating on different rows of the same bank, inserting new commands between write and read commands to the same chip select and allowing continuous read/write commands to execute before any command of opposite type. An option to disable the re-ordering logic can be implemented to allow the commands to execute in the order in which they initially entered the queue.
After re-ordering, the commands in the command queue can be further processed to maximize the data throughput by taking into consideration the memory bank readiness, availability of at-least one burst of data for write and read operations, bus turnaround timing and various conflicts among commands. An ageing scheme can be implemented to ensure that no low priority command remains in the queue indefinitely by elevating its priority after a programmable time interval.

**Memory PHY Interface**

Although a customized PHY interface for connecting the controller to the memory PHY can be implemented, the use of DFI compliant interface is recommended. A DFI compliant interface provides standardization of controller and PHY interface; thus reducing cost and facilitating re-use. It also reduces integration work specific to a vendor and provides seamless migration from an FPGA prototype to an ASIC. The other salient features of DFI compliance can be found out by referring the DFI specification.

**Memory PHY**

This module is connected to the memory system through ASIC pads and it drives data transactions onto it as per DDR3 electrical and protocol specifications. Apart from containing necessary logic for controlling DDR3 interface signals, it can have features like automatic write and read leveling, DLLs for timing control, capability of taking the memory in and out of various low power modes etc. Due to the limited scope of this paper, additional details on memory PHY can be obtained by referring any vendor datasheet.

**Pad Calibration**

The DDR3 memory controller can also have ASIC pad calibration logic to maintain proper voltage levels on the memory interface. This logic can be implemented like Off-Chip Driver calibration feature of DDR2 memories where a pull-up and pull-down legs of a resistor network is balanced to maintain proper voltage levels on memory interface across variations in operating conditions.

**Debugging Memory Controller**

The following additions can be implemented for assisting the debug process.

1. A number of status registers can be implemented in the controller providing an insight into the current controller and command execution states. Interrupts can also be incorporated to signal the occurrence of any critical event to the external system.
2. Observation registers can be added to the memory PHY which indicate the state of DLLs. This will help in the identification of any timing related issues.
3. A loopback provision can be implemented in the memory PHY which will help in testing proper connectivity and data eye generation without interfacing any memory component or programming the whole DDR3 controller.
4. ECC can be implemented in the controller to detect and correct any memory data corruptions.

**Conclusion**

In a nutshell, it can be clearly seen that DDR3 memories offer significant performance advantages compared to the DDR2 memories; while mandating changes to existing DDR2 memory connection topology and controller feature-set. Currently the DDR3 standard is superseded by DDR4 standard which promises even more efficient operation by reducing the operating voltage and increasing the frequency of memory clock. DDR4 memories have a typical operating voltage of 1.2V, supporting memory clock frequencies from 667MHz- 1.6GHz and memory densities up to 16Gb (which is limited to 8Gb for DDR3 memories); thus offering even higher performance at improved energy economy. While DDR4 is the latest and greatest of the DDR memory standards, DDR3 memories are still being widely used due to the current lower adoption rate of DDR4.
References

1. JEDEC DDR3 SDRAM standard (JESD79-3F)
2. JEDEC DDR2 SDRAM standard (JESD79-2F)

For more about the authors, visit the profiles for Biswaprakash Navajeevan and Vivek Singh.

If you liked this and would like to see a weekly collection of related products and features delivered directly to your inbox, click here to sign up for the EDN on Systems Design newsletter.