Introduction

Memory architectures are shifting from stub bus technology to high-speed linking. The traditional stub bus works well for slower devices, but has several limitations when supporting the higher bandwidth signals required in memory systems today. For instance, in a typical single-channel, multiple-slot registered dual in-line memory module (RDIMM) system, the topology of a data signal may include long motherboard traces on multiple PCB layers, short stubs between each DIMM socket, and at least one short stub on each module (see Figure 1).

Figure 1: RDIMM Single-Channel Data Line Board Topology
As the signal propagates from the memory controller through the connector and expands out to the DRAM, several of the traces can be mismatched. At slower clock rates this may not make a difference, but at higher signal rates, these signal mismatches can degrade signal quality. Other issues that can compromise RDIMM performance are:

- Signal integrity problems due to unbalanced signal loads from a mixture of different module configurations
- Limitations of bandwidth due to possible bus collisions

Bus contention can be characterized in a typical RDIMM system or directly from the DDR2 component data sheet. For example, in most RDIMM single-channel systems there is only one copy of data signals. This means all DRAM in any given bytelane will share a single board-level trace for their combined DQS signals. For the memory controller to query data from different slots or module ranks, it must ensure that there is not bus contention on the strobe lines. To do this, the controller must insert at least one dead clock cycle between sequential READ commands of the independent ranks. Furthermore, in a single-channel RDIMM system, it is impossible to simultaneously write data to any one rank while reading data from any other rank (regardless of how many RDIMMs are in the channel).

RDIMM systems have proven to be a stable and an excellent solution for multiple memory technologies. PC2-4200 RDIMM systems, which run with a data transfer rate of 533 MT/s per bit, can typically only support two dual-rank (DR) RDIMMs per memory channel. This is due to the stub bus architecture and the reflections this architecture causes. If more than two DR PC2-4200 (or faster) modules are required, the system may need to support a dual-channel design. This is a very limiting factor for systems that require extremely high memory density. This is because a dual-channel design duplicates all address, command, control, and data signals from the memory controller by routing them in parallel with the first channel. In some cases, a dual-channel design also requires a second memory controller and a large amount of board routing space.

Fully buffered DIMM (FBDIMM) systems offer virtually unlimited scalability of density, a significantly reduced number of routed motherboard signals, and high bandwidth solutions, all with an extremely reliable channel protocol. FBDIMM systems use DDR2 memory, but have a topology that uses a high-speed point-to-point interface between the controller and the first DIMM and between each other DIMM. The on-module interface between the advanced memory buffer (AMB) and the DDR2 DRAM completely isolates the DRAM from the high-speed channel and supports a point-to-two-point memory interface (see Figure 2). This new FBDIMM architecture also supports simultaneous READ and WRITE cycles within a single memory channel but on different DIMMs. For a comparison of a typical RDIMM channel to a typical FBDIMM channel, see Figure 3.

This technical note provides an introduction to the high-speed link, explains what to expect in regard to channel bandwidth, and outlines how to optimize performance, including some power analysis techniques.
Figure 2: FBDIMM Single-Channel Board Topology
Routing of one southbound bit
**Figure 3: Comparison of RDIMM Channel to FBDIMM Channel**

**RDIMM Channel (PC2-4200)**
- Peak channel bandwidth limited to peak DRAM bandwidth
  - Sustained channel bandwidth ~65% DRAM peak bandwidth
  - Cannot READ and WRITE simultaneously
  - *See Appendix for estimation of sustained DRAM bandwidth*

- ~150 active signals
- Memory controller → Maximum density = 2 dual-rank x4 RDIMMs

**FBDIMM Channel (PC2-5300)**
- Channel bandwidth = 1.5x of DRAM peak bandwidth or 8 GB/s
  - Can perform simultaneous READs and WRITEs

- Virtually unlimited density = Up to 8 dual-rank FBDIMMs
- *Can achieve high density with x4 or x8 DRAM*

- Northbound link (14 pairs)
- Southbound link (10 pairs)
FBDIMM Architecture

At the core of the FBDIMM is the AMB, which provides an interface from the DRAM to the high-speed channel (see Figure 4). Unlike previous memory module architectures, the AMB completely buffers the DRAM interface from the module edge connector. By isolating the DRAM from the high-speed channel, the DRAM can run independently of the other modules in the channel. This means that DRAM in different sockets, yet within the same channel, can run simultaneous tasks. For instance, the DRAM in socket 1 could perform a READ while the DRAM in socket 2 could simultaneously complete WRITE cycles. By isolating the DRAM from the high-speed bus, module density is not limited by signal fan-out or the capacitive loading of the additional DRAM. This isolation also keeps the entire DRAM interface local to the FBDIMM, where the signal integrity of all the DRAM signals is optimized.

Figure 4: FBDIMM Channel Block Diagram

For configuration and testing purposes, the AMB also supports a low-speed interface through the SMBus. The SMBus provides the memory controller access to the AMB registers with special debug and test modes. The SMBus also provides the unique address for each AMB within the channel.

The southbound channel consists of 10 differential pairs, or bitlanes, which carry command packets or WRITE data packets to the FBDIMMs. The northbound channel may include up to 14 bitlanes. The northbound channel carries the READ data packets from the FBDIMM to the memory controller. In case of minor high-speed signal problems, both the northbound and southbound channels have a fail-over feature. If the system detects a fault, the redundant bitlanes are remapped. During normal operation, the fail-over feature allows the high-speed link to optimize the redundant bitlanes for increased throughput.
The maximum possible channel bandwidth is defined by the reference clock. The reference clock is half the frequency of the DDR2 SDRAM clock and the high-speed data rate is 12 times that of the reference clock. The high-speed channel is designed to support twice as many READs as WRITEs. This combination provides a total channel bandwidth that is 50% greater than the actual DRAM peak. For example, if DDR2-667 memory is being used, a single FBDIMM channel can support a peak bandwidth of 8 GB/s. (A 64-bit DDR2-667 DRAM bus has a peak transfer rate of 5.3 GB/s.)

Southbound Channel

The southbound channel uses high-speed, point-to-point signals flowing from the memory controller to the AMB on the first FBDIMM. If there is more than one FBDIMM in the channel, the AMB on the first FBDIMM redrives the high-speed, point-to-point signals directly to the second FBDIMM in the channel. This point-to-point link between each FBDIMM continues for up to eight FBDIMMs in a single channel.

A southbound frame is made up of 12 transfers of 10 bits per transfer. There are two types of southbound frames: a command frame and a command-with-data frame. The command frame (see Figure 5) includes three commands with cyclic redundancy check (CRC) bits. A command frame can also include a combination of commands and partial WRITE data. To provide maximum flexibility in scheduling or queuing up the pipeline, the commands in the southbound frame can be directed to different FBDIMMs in the channel. For transfers of large packets of WRITE data, the southbound channel supports the command-with-data frame, which includes only one command and 72 bits of WRITE data, all with CRC bits (see Figure 6). Each frame is coded for a unique slot (or AMB). Additionally, as part of the high-speed protocol, the controller periodically sends out a sync command, which is required within every 42 clock cycles. This sync command is used to check the status of each AMB and to dynamically initialize clock synchronization.
Figure 5: Southbound – Command Frame with Full 10 Bitlanes Active

<table>
<thead>
<tr>
<th>Bit 9</th>
<th>Bit 8</th>
<th>Bit 7</th>
<th>Bit 6</th>
<th>Bit 5</th>
<th>Bit 4</th>
<th>Bit 3</th>
<th>Bit 2</th>
<th>Bit 1</th>
<th>Bit 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transfer 0</td>
<td>CRC</td>
<td>CRC</td>
<td>CRC</td>
<td>FT</td>
<td>CMD A</td>
<td>CMD A</td>
<td>CMD A</td>
<td>CMD A</td>
<td>CMD A</td>
</tr>
<tr>
<td>Transfer 1</td>
<td>CRC</td>
<td>CRC</td>
<td>CRC</td>
<td>FT</td>
<td>CMD A</td>
<td>CMD A</td>
<td>CMD A</td>
<td>CMD A</td>
<td>CMD A</td>
</tr>
<tr>
<td>Transfer 2</td>
<td>CRC</td>
<td>CRC</td>
<td>CRC</td>
<td>CRC</td>
<td>CMD A</td>
<td>CMD A</td>
<td>CMD A</td>
<td>CMD A</td>
<td>CMD A</td>
</tr>
<tr>
<td>Transfer 3</td>
<td>CRC</td>
<td>CRC</td>
<td>CRC</td>
<td>CRC</td>
<td>CMD A</td>
<td>CMD A</td>
<td>CMD A</td>
<td>CMD A</td>
<td>CMD A</td>
</tr>
<tr>
<td>Transfer 4</td>
<td>CRC</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>CMD B</td>
<td>CMD B</td>
<td>CMD B</td>
<td>CMD B</td>
<td>CMD B</td>
</tr>
<tr>
<td>Transfer 5</td>
<td>CRC</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>CMD B</td>
<td>CMD B</td>
<td>CMD B</td>
<td>CMD B</td>
<td>CMD B</td>
</tr>
<tr>
<td>Transfer 6</td>
<td>CRC</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>CMD B</td>
<td>CMD B</td>
<td>CMD B</td>
<td>CMD B</td>
<td>CMD B</td>
</tr>
<tr>
<td>Transfer 7</td>
<td>CRC</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>CMD B</td>
<td>CMD B</td>
<td>CMD B</td>
<td>CMD B</td>
<td>CMD B</td>
</tr>
<tr>
<td>Transfer 8</td>
<td>CRC</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>CMD C</td>
<td>CMD C</td>
<td>CMD C</td>
<td>CMD C</td>
<td>CMD C</td>
</tr>
<tr>
<td>Transfer 9</td>
<td>CRC</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>CMD C</td>
<td>CMD C</td>
<td>CMD C</td>
<td>CMD C</td>
<td>CMD C</td>
</tr>
<tr>
<td>Transfer 10</td>
<td>CRC</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>CMD C</td>
<td>CMD C</td>
<td>CMD C</td>
<td>CMD C</td>
<td>CMD C</td>
</tr>
<tr>
<td>Transfer 11</td>
<td>CRC</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>CMD C</td>
<td>CMD C</td>
<td>CMD C</td>
<td>CMD C</td>
<td>CMD C</td>
</tr>
</tbody>
</table>

CRC = Cyclic redundancy check bits used for command A and frame type only
FT = Frame type; identifies the frame as a command, data, or other type
CMD X = Command A, B, or C; each command includes 24 coded bits
CRC = Cyclic redundancy check bits used for commands B and C; 14 additional CRC bits are coded within the next southbound frame
0 = Unused or reserved bits

Figure 6: Southbound – Command-with-Data Frame with Full 10 Bitlanes Active

<table>
<thead>
<tr>
<th>Bit 9</th>
<th>Bit 8</th>
<th>Bit 7</th>
<th>Bit 6</th>
<th>Bit 5</th>
<th>Bit 4</th>
<th>Bit 3</th>
<th>Bit 2</th>
<th>Bit 1</th>
<th>Bit 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transfer 0</td>
<td>CRC</td>
<td>CRC</td>
<td>CRC</td>
<td>FT</td>
<td>CMD A</td>
<td>CMD A</td>
<td>CMD A</td>
<td>CMD A</td>
<td>CMD A</td>
</tr>
<tr>
<td>Transfer 1</td>
<td>CRC</td>
<td>CRC</td>
<td>CRC</td>
<td>FT</td>
<td>CMD A</td>
<td>CMD A</td>
<td>CMD A</td>
<td>CMD A</td>
<td>CMD A</td>
</tr>
<tr>
<td>Transfer 2</td>
<td>CRC</td>
<td>CRC</td>
<td>CRC</td>
<td>CRC</td>
<td>CMD A</td>
<td>CMD A</td>
<td>CMD A</td>
<td>CMD A</td>
<td>CMD A</td>
</tr>
<tr>
<td>Transfer 3</td>
<td>CRC</td>
<td>CRC</td>
<td>CRC</td>
<td>CRC</td>
<td>CMD A</td>
<td>CMD A</td>
<td>CMD A</td>
<td>CMD A</td>
<td>CMD A</td>
</tr>
<tr>
<td>Transfer 4</td>
<td>CRC</td>
<td>W_DATA</td>
<td>W_DATA</td>
<td>W_DATA</td>
<td>W_DATA</td>
<td>W_DATA</td>
<td>W_DATA</td>
<td>W_DATA</td>
<td>W_DATA</td>
</tr>
<tr>
<td>Transfer 5</td>
<td>CRC</td>
<td>W_DATA</td>
<td>W_DATA</td>
<td>W_DATA</td>
<td>W_DATA</td>
<td>W_DATA</td>
<td>W_DATA</td>
<td>W_DATA</td>
<td>W_DATA</td>
</tr>
<tr>
<td>Transfer 6</td>
<td>CRC</td>
<td>W_DATA</td>
<td>W_DATA</td>
<td>W_DATA</td>
<td>W_DATA</td>
<td>W_DATA</td>
<td>W_DATA</td>
<td>W_DATA</td>
<td>W_DATA</td>
</tr>
<tr>
<td>Transfer 7</td>
<td>CRC</td>
<td>W_DATA</td>
<td>W_DATA</td>
<td>W_DATA</td>
<td>W_DATA</td>
<td>W_DATA</td>
<td>W_DATA</td>
<td>W_DATA</td>
<td>W_DATA</td>
</tr>
<tr>
<td>Transfer 8</td>
<td>CRC</td>
<td>W_DATA</td>
<td>W_DATA</td>
<td>W_DATA</td>
<td>W_DATA</td>
<td>W_DATA</td>
<td>W_DATA</td>
<td>W_DATA</td>
<td>W_DATA</td>
</tr>
<tr>
<td>Transfer 9</td>
<td>CRC</td>
<td>W_DATA</td>
<td>W_DATA</td>
<td>W_DATA</td>
<td>W_DATA</td>
<td>W_DATA</td>
<td>W_DATA</td>
<td>W_DATA</td>
<td>W_DATA</td>
</tr>
<tr>
<td>Transfer 10</td>
<td>CRC</td>
<td>W_DATA</td>
<td>W_DATA</td>
<td>W_DATA</td>
<td>W_DATA</td>
<td>W_DATA</td>
<td>W_DATA</td>
<td>W_DATA</td>
<td>W_DATA</td>
</tr>
<tr>
<td>Transfer 11</td>
<td>CRC</td>
<td>W_DATA</td>
<td>W_DATA</td>
<td>W_DATA</td>
<td>W_DATA</td>
<td>W_DATA</td>
<td>W_DATA</td>
<td>W_DATA</td>
<td>W_DATA</td>
</tr>
</tbody>
</table>

CRC = Cyclic redundancy check bits used for command A and frame type; these CRC bits are XOR-ed with the leftover 14 CRC bits from the last southbound frame.
FT = Frame type; identifies the frame type and slot to which the data belongs
CMD A = Command A; includes 24 coded bits
CRC = Cyclic redundancy check bits used for the data and ECC bits; 14 additional CRC bits are coded within the next southbound frame
W_DATA = 64 bits of WRITE data and 8 bits of ECC data
Example: WRITE (BL=4) to One or More FBDIMMs in a Channel

A WRITE (burst length = 4) to one or more FBDIMMs within the channel may consist of at least one command frame combined with three command-with-data frames for each WRITE sequence (see Figure 7). In this example, the controller sends three ACTIVE commands in the first command frame. Command A is issued on the next DRAM cycle, where commands B and C are delayed by one additional cycle.

All three commands are within the single southbound frame. The AMB automatically inserts DESELECT commands to the DRAM devices when the bus is idle, but if a southbound frame is issued, it must contain a valid command. This is why there are NOP commands within the frames. Following the first command frame, there are several command-with-data frames; these include 72-bits of WRITE data within each cycle and a WRITE command to the FBDIMM in slot 1. After the frames are decoded by the respective AMB, the result at the DRAM is a normal string of commands and data.

This is just one of several ways to WRITE data to the DRAM. The AMB includes an integrated FIFO, and it can be used in creative ways to optimize the throughput of a given slot. Additionally, other commands can be issued instead of NOPs; NOPs were used to keep the example simple.

Theoretically, the southbound channel is intended to transfer WRITE data at half of the peak transfer rate of the DDR2 SDRAM. This means that, for DDR2-667, the southbound channel has a peak bandwidth of 2.67 GB/s. However, when combining the required DRAM commands—clock-sync frames with data transfers—the sustainable southbound data bandwidth will be slightly less.

Figure 7: Southbound Frames and the Related DRAM Decode (at each FBDIMM)
Northbound Channel

The northbound channel flows from the FBDIMM to the controller. Like the southbound channel, it is also point-to-point and is redriven between each AMB/FBDIMM and eventually between the first FBDIMM and the memory controller. The northbound channel also uses frames to transfer individual FBDIMM status and READ data from the FBDIMMs back to the controller. The four types of northbound frames are: data, status, idle, and alert. The northbound data frame is the most important in regards to channel bandwidth. The data frame can transfer up to two complete packets of 72-bit READ data, including CRC bits for all 144 bits of data. Although the other northbound frames provide important functions, their primary purpose is to monitor the status and reliability of the channel. This technical note does not provide additional detail about those frame types.

The northbound channel can support a maximum of 14 differential signal pairs; some system designs may only support 12. The level of northbound channel redundancy and CRC protection is defined by the number of signal pairs that are supported. If the channel supports 14 bitlanes, it is capable of full 12-bit CRC with 72-bits of data and has redundant signal pairs. If the channel only supports 12 bitlanes, the frame only supports a 6-bit CRC with 64 bits of data and no fail-over redundancy. The 14-bit data frame includes 144 bits of data with 2 bits of CRC code for every 12 bits of data in one transfer (see Figure 8).

The largest northbound frame consists of twelve 14-bit, high-speed transfers. Unlike the southbound frames, the northbound data frame only carries data and CRC bits and is not burdened by commands. The lower 12 bits contain the DRAM data and ECC data, which are mapped to individual DRAM devices.

Figure 8: Configuration of a Northbound 14-bitlane Data Frame

<table>
<thead>
<tr>
<th>Bit 13</th>
<th>Bit 12</th>
<th>Bit 11</th>
<th>Bit 10</th>
<th>Bit 9</th>
<th>Bit 8</th>
<th>Bit 7</th>
<th>Bit 6</th>
<th>Bit 5</th>
<th>Bit 4</th>
<th>Bit 3</th>
<th>Bit 2</th>
<th>Bit 1</th>
<th>Bit 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transfer 0</td>
<td>CRC</td>
<td>CRC</td>
<td>R_DATA</td>
<td>R_DATA</td>
<td>R_DATA</td>
<td>R_DATA</td>
<td>R_DATA</td>
<td>R_DATA</td>
<td>R_DATA</td>
<td>R_DATA</td>
<td>R_DATA</td>
<td>R_DATA</td>
<td>R_DATA</td>
</tr>
<tr>
<td>Transfer 1</td>
<td>CRC</td>
<td>CRC</td>
<td>R_DATA</td>
<td>R_DATA</td>
<td>R_DATA</td>
<td>R_DATA</td>
<td>R_DATA</td>
<td>R_DATA</td>
<td>R_DATA</td>
<td>R_DATA</td>
<td>R_DATA</td>
<td>R_DATA</td>
<td>R_DATA</td>
</tr>
<tr>
<td>Transfer 2</td>
<td>CRC</td>
<td>CRC</td>
<td>R_DATA</td>
<td>R_DATA</td>
<td>R_DATA</td>
<td>R_DATA</td>
<td>R_DATA</td>
<td>R_DATA</td>
<td>R_DATA</td>
<td>R_DATA</td>
<td>R_DATA</td>
<td>R_DATA</td>
<td>R_DATA</td>
</tr>
<tr>
<td>Transfer 3</td>
<td>CRC</td>
<td>CRC</td>
<td>R_DATA</td>
<td>R_DATA</td>
<td>R_DATA</td>
<td>R_DATA</td>
<td>R_DATA</td>
<td>R_DATA</td>
<td>R_DATA</td>
<td>R_DATA</td>
<td>R_DATA</td>
<td>R_DATA</td>
<td>R_DATA</td>
</tr>
<tr>
<td>Transfer 4</td>
<td>CRC</td>
<td>CRC</td>
<td>R_DATA</td>
<td>R_DATA</td>
<td>R_DATA</td>
<td>R_DATA</td>
<td>R_DATA</td>
<td>R_DATA</td>
<td>R_DATA</td>
<td>R_DATA</td>
<td>R_DATA</td>
<td>R_DATA</td>
<td>R_DATA</td>
</tr>
<tr>
<td>Transfer 5</td>
<td>CRC</td>
<td>CRC</td>
<td>R_DATA</td>
<td>R_DATA</td>
<td>R_DATA</td>
<td>R_DATA</td>
<td>R_DATA</td>
<td>R_DATA</td>
<td>R_DATA</td>
<td>R_DATA</td>
<td>R_DATA</td>
<td>R_DATA</td>
<td>R_DATA</td>
</tr>
<tr>
<td>Transfer 6</td>
<td>CRC</td>
<td>CRC</td>
<td>R_DATA</td>
<td>R_DATA</td>
<td>R_DATA</td>
<td>R_DATA</td>
<td>R_DATA</td>
<td>R_DATA</td>
<td>R_DATA</td>
<td>R_DATA</td>
<td>R_DATA</td>
<td>R_DATA</td>
<td>R_DATA</td>
</tr>
<tr>
<td>Transfer 7</td>
<td>CRC</td>
<td>CRC</td>
<td>R_DATA</td>
<td>R_DATA</td>
<td>R_DATA</td>
<td>R_DATA</td>
<td>R_DATA</td>
<td>R_DATA</td>
<td>R_DATA</td>
<td>R_DATA</td>
<td>R_DATA</td>
<td>R_DATA</td>
<td>R_DATA</td>
</tr>
<tr>
<td>Transfer 8</td>
<td>CRC</td>
<td>CRC</td>
<td>R_DATA</td>
<td>R_DATA</td>
<td>R_DATA</td>
<td>R_DATA</td>
<td>R_DATA</td>
<td>R_DATA</td>
<td>R_DATA</td>
<td>R_DATA</td>
<td>R_DATA</td>
<td>R_DATA</td>
<td>R_DATA</td>
</tr>
<tr>
<td>Transfer 9</td>
<td>CRC</td>
<td>CRC</td>
<td>R_DATA</td>
<td>R_DATA</td>
<td>R_DATA</td>
<td>R_DATA</td>
<td>R_DATA</td>
<td>R_DATA</td>
<td>R_DATA</td>
<td>R_DATA</td>
<td>R_DATA</td>
<td>R_DATA</td>
<td>R_DATA</td>
</tr>
<tr>
<td>Transfer 10</td>
<td>CRC</td>
<td>CRC</td>
<td>R_DATA</td>
<td>R_DATA</td>
<td>R_DATA</td>
<td>R_DATA</td>
<td>R_DATA</td>
<td>R_DATA</td>
<td>R_DATA</td>
<td>R_DATA</td>
<td>R_DATA</td>
<td>R_DATA</td>
<td>R_DATA</td>
</tr>
<tr>
<td>Transfer 11</td>
<td>CRC</td>
<td>CRC</td>
<td>R_DATA</td>
<td>R_DATA</td>
<td>R_DATA</td>
<td>R_DATA</td>
<td>R_DATA</td>
<td>R_DATA</td>
<td>R_DATA</td>
<td>R_DATA</td>
<td>R_DATA</td>
<td>R_DATA</td>
<td>R_DATA</td>
</tr>
</tbody>
</table>

CRC = Cyclic redundancy check bits
R_DATA = 72-bits of READ and ECC data, 1st transfer
R_DATA = 72-bits of READ and ECC data, 2nd transfer
The AMB supports fail-over mode so, when running with the full 14 bitlanes, there are 12 CRC bits. If a problem is detected, the frame drops one bitlane, remaps the data, and runs with 6 CRC bits. The 13-bitlane data frame has the same format as the 14-bitlane data frame, except it has one fewer CRC bit per transfer. In the worst case, the channel can run with a 12-bitlane data frame. In this case, all CRC bits and the 8 ECC bits are eliminated.

The DRAM receives all commands from the southbound frames, so the DRAM timing for the READs looks very similar to the timing for the WRITEs. For example, Figure 9 shows how simple it is to perform sequential READs from different FBDIMMs within the channel. Within the AMB, there is at least a one-cycle delay from the time data is received from the DRAM to posting it to the high-speed bus (northbound channel).

**Figure 9: Northbound Frames and the Related DRAM Decode at each FBDIMM**
Channel Utilization

The northbound channel is optimized for a nearly continuous throughput of READ data. However, unless the system is doing continuous READs from the DRAM, or there is more than one FBDIMM in the channel, it is unrealistic to expect the sustained northbound channel bandwidth to match that of the peak DRAM bandwidth. This is due to the same DRAM timing limitations that appear in a single RDIMM channel. In an FBDIMM channel, the DRAM timing limitations are on the secondary side of the AMB, or are isolated to each FBDIMM. This means that by increasing the number of FBDIMMs in the channel, the DRAM timing limitations can be overcome.

The peak bandwidth of a DDR2-667 64-bit bus is approximately 5.3 GB/s (667 MT/s per bit multiplied by 64 bits, divided by 8). However, due to various DDR2 timing limitations, the typical sustained bandwidth of the 64-bit DRAM bus is roughly 3.4 GB/s (about 65% of peak). As such, even if a DDR2-667 FBDIMM channel is capable of running at 8 GB/s, and if only one FBDIMM is installed, it is limited to the maximum sustained throughput of that single FBDIMM—about 3.4 GB/s (see Figure 10).

Figure 10: Limited Bandwidth of the High-Speed Channel with a Single FBDIMM

Notes:
1. Even though the FBDIMM high-speed bus can support 1.5x the DRAM peak bandwidth, with only one FBDIMM installed, it is limited to the maximum bandwidth of that single FBDIMM.
2. As with an RDIMM, a single FBDIMM can sustain about 65% of the peak DRAM bandwidth (~3.4 GB/s for DDR2-667).
One way of overcoming this bandwidth limitation and achieving the maximum potential of an FBDIMM system is to use at least two FBDIMMs (see Figure 11). An evaluation of the same DDR2-667 FBDIMM channel with two FBDIMMs installed has shown it can achieve a sustained bandwidth of about 6.8 GB/s (3.4 GB/s from each FBDIMM). This is nearly 30% more sustained bandwidth than the peak bandwidth (5.3 GB/s) of a single DDR2-667 RDIMM channel. Additionally, if the same FBDIMM channel were populated with three or more modules, the theoretical sustained bandwidth could be 5.3 GB/s for READs only, or the full 8 GB/s if both READs and WRITEs were performed.

Although system-dependent, typical FBDIMM systems, when evaluated, show that the total channel bandwidth is a sum of its parts. Additionally, the number of data transactions are usually split evenly between all slots and ranks within the system with the first slot getting slightly more hits. This makes it easy to estimate the total channel bandwidth and percentage of DRAM usage.

**Figure 11: Sustaining Maximum FBDIMM Bandwidth – Sum of the Parts**

* Channel bandwidth is limited by the combined DRAM bandwidth of 6.8 GB/s (3.4 GB/s + 3.4 GB/s), which is less than the capable channel bandwidth of 8 GB/s.

* Maximum channel bandwidth of 8 GB/s is achieved with three or more FBDIMMs installed. Due to channel saturation, with three or more FBDIMMs installed, the channel may limit the DRAM bandwidth.

* Assumes sustained DRAM utilization of 3.4 GB/s (about 65%)

Channel bandwidth = (FBDIMM bandwidth) x (number of FBIMMs in channel or peak channel bandwidth, whichever is less)
Scalable Power

The way the FBDIMM channel is populated can play an important role in channel bandwidth and performance. Performance translates to power consumption. When designing an FBDIMM system, it is important to balance optimal bandwidth with minimal power impact. Most FBDIMM systems can be made scalable to bandwidth, power, and density requirements. Like previous memory technologies, there are several FBDIMM configurations available (see Table 1).

### Table 1: FBDIMM Configurations

<table>
<thead>
<tr>
<th>Type</th>
<th>Configuration</th>
<th>Number of DRAM</th>
<th>ECC Supported</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single rank</td>
<td>(SR x8)</td>
<td>9 die</td>
<td>Yes</td>
</tr>
<tr>
<td>Single rank</td>
<td>(SR x4)</td>
<td>18 die</td>
<td>Yes</td>
</tr>
<tr>
<td>Dual rank</td>
<td>(DR x8)</td>
<td>18 die</td>
<td>Yes</td>
</tr>
<tr>
<td>Dual rank</td>
<td>(DR x4)</td>
<td>36 die</td>
<td>Yes</td>
</tr>
</tbody>
</table>

There are several ways to design a high-density FBDIMM system. A system designer can provide up to eight slots per channel. This provides an easy upgrade path, particularly if high density may not be needed immediately, or if the designer wants to build the density using less-expensive, lower-density FBDIMMs. However, this solution could saturate the high-speed bus and limit the bandwidth of the individual FBDIMMs when the system is fully populated. Likewise, because the throughput of the channel is divided between the individual loads, this method may have the best per-slot power efficiency. For example, if the total channel supports 8 GB/s, and there are eight FBDIMMs installed in the channel, each slot would provide an average of 1 GB/s. With a bandwidth of 1 GB/s, the power consumption per FBDIMM would be significantly lower than if the 8 GB/s were distributed between only three or four FBDIMMs.

As with previous module technologies, there is a substantial difference in power consumption between x4- and x8-based FBDIMMs. This is primarily due to the number of active DRAM devices on each module (x4-based modules include twice as many DRAM devices). Due to the increased number of available slots in an FBDIMM channel, x8-based modules are more popular (see Figure 12).
Many systems try to maximize bandwidth, limit power consumption per channel, and maintain system flexibility. Due to the lower number of signals on the northbound and southbound channels, it is easier to design a multiple-channel FBDIMM system. This enables the system designer to tune the channel for high throughput without saturating the high-speed bus. It also provides the ability to electrically and physically interleave the channels to best accommodate thermal conditions. As discussed previously, both the southbound and northbound channels are redriven by the AMB. The last AMB in the channel does not need to redrive the southbound channel, nor does it need to receive upstream northbound signals. This means that the last FBDIMM in the channel typically consumes less AMB power (on the 1.5V rail) than an AMB that is in series within the channel (see Figure 13).

**Figure 13: Effect of Slot Position on AMB Power**

<table>
<thead>
<tr>
<th>Power per AMB</th>
<th>AMB(1)</th>
<th>AMB(2)</th>
<th>AMB(3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.2W</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.4W</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.6W</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.8W</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.0W</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.2W</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.4W</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.6W</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.8W</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

AMB(1), AMB(2) and AMB(3) consist of different vendors and/or lots.
General FBDIMM Power and Thermal Characteristics

To ensure a robust and reliable design, the FBDIMM system requires adequate cooling. Cooling requirements are directly related to the power dissipation of the FBDIMM system. Micron offers many tools to help determine the cooling requirements, including our exclusive DDR2 power calculator and an FBDIMM thermal design guide.

The AMB includes an on-board thermal sensor that provides a real-time AMB temperature to the memory controller. This allows the memory controller to monitor the AMB and invoke power throttling in extreme temperature conditions. Through power throttling, the memory controller may limit the DRAM bandwidth by slowing down memory accesses. The slower the bandwidth, the less power the DRAM consumes. Additionally, all Micron FBDIMMs are shipped with a full module heat spreader (FMHS) to help dissipate the high temperatures of the AMB (see Figure 14).

Figure 14: Micron FBDIMM with Full Module Heat Spreader

Most FBDIMM system designers perform a complete thermal simulation of the high-speed channel. This determines the amount of airflow required to reliably operate the FBDIMMs under all circumstances. Each system is different, and the amount of airflow depends on the inlet temperature of the air. However, most systems require airflow of at least 3m/s. To ensure sufficient cooling, Micron recommends completing a full thermal simulation for each FBDIMM system design (see Figure 15).

Figure 15: FBDIMM Thermal Simulation Results

Before engineers can perform a proper system thermal simulation, they must know the expected power per device, per slot, and per system. Because the FBDIMM system is scalable, it is easy to estimate the system power consumption. FBDIMM power is calculated using the sustained channel bandwidth, the configuration of FBDIMMs, and the number of FBDIMMs that will be installed in the completed system.
Estimating Bandwidth

The first step in estimating FBDIMM power is to estimate the bandwidth. The individual rank bandwidth is either the sustained DRAM bandwidth or a percentage of the total channel bandwidth, whichever is less. To estimate the bandwidth of an individual rank, the sustainable channel and DRAM bandwidth must be determined:
1. Determine the bandwidth of each FBDIMM slot.
2. Calculate the bandwidth of each rank.
3. Estimate the percent of use for each rank.
4. Find the percent of READs and WRITEs to which each rank contributes.

Note: The sustained DRAM bandwidth is system dependent but, in most circumstances, it will not exceed about 65% of the peak DRAM bandwidth. The channel bandwidth is also system dependent but can never be more than 1.5 times the peak DRAM bandwidth.

Example 1: Estimating Individual FBDIMM Bandwidth (total channel bandwidth is known)

FBDIMM bandwidth = (channel bandwidth / number of FBDIMMs in the channel)  
(EQ 1)

Notes:  
1. Individual FBDIMM bandwidth must be equal to or less than the sustained DRAM utilization.

• An FBDIMM channel has four slots, but only two slots are populated. Within each of these two slots is a single DR FBDIMM.
• The total channel sustained bandwidth = 6 GB/s at DDR2-667.
• Bandwidth per slot = (6 GB/s) divided by (2 slots) = 3 GB/s per slot, which is less than the sustained DRAM utilization of 3.4 GB/s.

Example 2: Estimating Total Channel Bandwidth (FBDIMM bandwidth is known)

Total channel bandwidth = (sustained FBDIMM bandwidth x number of FBDIMMs in channel)  
(EQ 2)

Notes:  
1. Total channel bandwidth must be equal to or less than 8 GB/s for DDR2-667.

• Peak channel bandwidth for DDR2-667 is 8 GB/s.
• FBDIMM channel has four slots and each slot is populated with a DR FBDIMM.
• The sustained FBDIMM bandwidth is 3.4 GB/s (about 65% of peak DRAM throughput).

Total channel bandwidth = (4 slots) x (3.4 GB/s) = 13.6 GB/s, but the peak channel bandwidth is limited to 8 GB/s, so the sustained channel bandwidth is 8 GB/s. This makes the peak FBDIMM bandwidth equal to 3.4 GB/s but the sustained FBDIMM bandwidth equal to 2 GB/s (see Example 1: Estimating Individual FBDIMM Bandwidth (total channel bandwidth is known)).
To estimate individual rank bandwidth, divide the channel bandwidth equally between all FBDIMMs installed in the system. If this is less than or equal to the sustained DRAM bandwidth, this is the bandwidth for each slot.

If it is more than the sustained DRAM bandwidth, use the sustained DRAM bandwidth for each slot. If the system is using single-rank (SR) FBDIMMs, this is the average bandwidth for each rank in the system. If the system is using DR FBDIMMs, divide the individual slot bandwidth by two.

**Example 3: Estimating Individual Rank Bandwidth (sustained FBDIMM bandwidth is known)**

\[
\text{Rank bandwidth} = \left(\text{individual FBDIMM bandwidth} / \text{number of ranks on FBDIMM}\right) \quad (\text{EQ 3})
\]

- Individual FBDIMM bandwidth (from Example 1: Estimating Individual FBDIMM Bandwidth (total channel bandwidth is known)) = 3 GB/s.
- FBDIMM modules are dual rank.
- The average individual-rank bandwidth is (3 GB/s) divided by (2 ranks) = 1.5 GB/s.

Once the bandwidth per rank is determined, the individual DRAM power can be estimated using the Micron DDR2 power calculator. Before using the power calculator, the percentage of DRAM READs and WRITEs need to be determined. For FBDIMMs, there is a 2:1 ratio of READs to WRITEs. Start with the known sustained individual bandwidth per rank and divide by the peak DRAM bandwidth. Then apply the 2:1 ratio to the percentage of READs to WRITEs.

**Example 4: Estimating the Percent of Bandwidth per Rank (individual rank bandwidth is known)**

\[
\text{Percent of bandwidth per rank} = \left(\text{rank bandwidth} / \text{peak DRAM bandwidth}\right) \quad (\text{EQ 4})
\]

- Individual sustained rank bandwidth (from Example 3: Estimating Individual Rank Bandwidth (sustained FBDIMM bandwidth is known)) = 1.5 GB/s.
- Peak DRAM bandwidth at DDR2-667 = 5.33 GB/s.
- Percent of total bandwidth per rank = (1.5 GB/s) divided by (5.33 GB/s) = 28%.
Example 5: Estimating the Percent of READs (percent of rank bandwidth is known)

\[
\text{Percent of READs per rank} = (\text{rank bandwidth} \times 66\%) \tag{EQ 5}
\]

- Total sustained rank percent is 28% (from Example 4: Estimating the Percent of Bandwidth per Rank (individual rank bandwidth is known)).
- The approximate percent of READs = (66%) \times (28\% \text{ total rank bandwidth}) = 18.5\%.

Example 6: Estimating Percent of WRITEs (percent of rank bandwidth is known)

\[
\text{Percent of WRITEs per rank} = (\text{rank bandwidth} \times 33\%) \tag{EQ 6}
\]

- Total sustained rank percent is 28% (from Example 4: Estimating the Percent of Bandwidth per Rank (individual rank bandwidth is known)).
- The approximate percent of WRITEs = (33\%) \times (28\% \text{ total rank bandwidth}) = 9.25\%.

In Examples 2–6, the channel can sustain a total bandwidth of 6 GB/s. The channel has two DR FBDIMMs installed, so each FBDIMM is running at 3 GB/s, which makes each individual rank run about 1.5GB/s (28% of the peak DRAM bandwidth).
- Four-slot channel with only two FBDIMMs installed
- Each FBDIMM is dual-rank
- Total channel sustained bandwidth = 6 GB/s
- Individual FBDIMM bandwidth = 3 GB/s
- Bandwidth per rank = 1.5 GB/s
- Each DRAM has about 18.5% READs and 9.25% WRITEs
After the actual rank READ/WRITE percentages have been determined, the per-DRAM current and power can be estimated using the Micron DDR2 power calculator with the appropriate device IDD values. It must also be determined if the system is using a burst length (BL) of 4 or 8. Using the values from Example 6: Estimating Percent of WRITEs (percent of rank bandwidth is known), with BL = 4, and using data sheet values for our 1Gb DDR2-667 DRAM, each DRAM’s power is predicted to be about 280mW (see Figure 16).

Figure 16:  Estimated Individual DRAM Power for 1.5 GB/s (with 28% DRAM throughput)

The estimated FBDIMM power is simply the individual DRAM value from Figure 16 multiplied by the number of DRAM devices on the FBDIMM, plus the AMB power. For example, for two DR x8 FBDIMMs in a system with a total channel bandwidth of 6 GB/s and with AMB power of 6W, the approximate individual FBDIMM power is about 11W.

- Each FBDIMM has two ranks with nine DRAM per rank.
- Each DRAM consumes about 280mW of power.
- The AMB consumes about 6W of power.
- Total FBDIMM power = (18 x 280mW) + 6W ≈ 11W

Note: AMB power can vary by vendor, operating condition, and slot position, so this technical note does not try to predict the exact AMB power. Contact the AMB vendor or Micron directly for additional estimated AMB power values.
Summary

The new FBDIMM architecture offers features including extraordinarily high bandwidth, virtually unlimited channel density, scalable performance, and compatibility between modules. Unlike a single RDIMM channel, a single FBDIMM channel can sustain 1.5 times the peak DRAM bandwidth. At DDR2-667 speeds, this equates to a single FBDIMM channel peak bandwidth of 8 GB/s and a dual-channel FBDIMM system with 16 GB/s. With optimized system software and proper channel design, it is possible to sustain these high bandwidths. Additionally, due to the point-to-point nature of the AMB, there is no loading penalty for adding slots in the channel. In fact, a single FBDIMM channel can support up to eight individual slots. With the ability to populate up to eight slots in any single channel, FBDIMM systems offer unsurpassed scalability and extended channel density.

This technical note outlines the following steps for estimating bandwidth and power for FBDIMMs:
1. Estimate the bandwidth per rank
2. Determine the percentage of DRAM READs/WRITEs
3. Use the Micron DDR2 power calculator
4. Add in the AMB power
Appendix

Estimating Sustained DRAM Bandwidth

Many times, the sustained bandwidth of a DRAM device is less than the absolute peak bandwidth. This depends on how the device is used, the speed grade, and various data sheet timing parameters. For example, \( t_{RRD} \) (MIN) can affect the sustained bandwidth of a DDR2-667 device, as follows:

**Number of Command Sets**

\[
\text{Number of command sets} = \left( \frac{t_{RC}}{\text{command sets}} \right) \quad (\text{EQ 7})
\]

**Command Set**

\[
\text{Command set} = (\text{ACTIVE + READ}) \text{ or } (\text{ACTIVE + WRITE}) = 2 \text{ clocks} \quad (\text{EQ 8})
\]

**For DDR2-667 (-3E speed grade)**

- Minimum clock cycle time \( t_{CK} = 3\,\text{ns} \) (1 clock)
- Minimum time to open/close a single bank \( t_{RC} = 57\,\text{ns} \) (19 clocks)
- Minimum time between activate commands \( t_{RRD} = 7.5\,\text{ns} \) (3 clocks)
- Assuming minimum burst length BL = 4 (2 clocks)

At 100% DRAM utilization, there would not be any NOP or DESELECT commands within any \( t_{RC} \) time period. Using Equation 7, the maximum number of command sets possible when running at 100% bandwidth can be computed. This assumes there are no timing violations or restrictions on the DRAM. At 100% DRAM bandwidth, there would be an average of 6ns (2 clock cycles) between ACTIVE commands.

**At 100% DRAM Bandwidth**

- Maximin number of command sets = \( t_{RC} \) / command sets
  - = 19 clocks / 2
  - = 9.5 command sets
- At 100% bandwidth there are 9.5 command sets per \( t_{RC} \).

However, with 9.5 command sets per \( t_{RC} \), there would have to be an active command an average of every 6ns (2 clocks). By specification, the minimum time between active commands to the same DRAM is \( t_{RRD} \) (7.5ns). Timing specs must be rounded up to whole clock cycles, so \( t_{RRD} \) (MIN) = 3 clocks.
Adjusted Percent of DRAM Bandwidth

Adjusted percent of DRAM bandwidth = \( \frac{t_{RC}}{t_{RRD}} \times \frac{100\% BW}{100\% BW} \) (EQ 9)

Due to the limitations of \( t_{RRD} \) (MIN), the adjusted percent of DRAM bandwidth, or the sustainable DRAM bandwidth, is about 67%:

- Adjusted percent bandwidth = \( \frac{19 \text{ clock cycles}}{3 \text{ clock cycles}} \times \frac{100\% BW}{9.5 \text{ clocks}} \)
- = 6.33 clocks / 9.50 clocks
- = 67%

This example only takes into account one DRAM timing parameter (\( t_{RRD} \)). If we add in others or look at other speed grades, the sustainable bandwidth could vary. Under heavy use, DRAM use is typically in the range of 60% to 70%.
Revision History

Rev. B .................................................................................................................. 12/09
  • Updated format
  • Minor grammatical changes

Rev. A .................................................................................................................. 10/06
  • Initial release