
In any memory system, the topology limits the internal architecture of the memory storage devices and the memory access protocol. As described in class, the topologies of the SDRAM based memory system and the RDRAM memory system each dictated the access protocol of the respective memory systems. For the support of modern processors, high bandwidth, low latency, and high throughput constraints dictate that a topology suitable for use with a packet based command and response memory system similar to the topology of the DRDRAM memory system is required. For this reason, we propose the use of a path length matched memory system whose topology is shown as figure 1.

In figure 1, we propose a radically different topology for a future memory system. In this figure, we show that the command packet, as well as the data for a write command, are synchronized with respect to the command clock. In contrast, data from a memory read request is sent by a memory chip in the system that sends the data in a packet format that is synchronized with respect to the data clock.

The advantage to a uni-directional topology is that it eliminates the bus turnaround time of the bi-directional data bus. Furthermore, by arranging the data channel in a manner such that the path lengths are perfectly matched with respect to each other, the round trip path length will be the same regardless of the location of the chip in the channel. The uniformity in topology and unidirectional nature means that the entire system may be able to operate at higher frequencies, and the wire delay latencies are static and uniform, making control logic possibly easier to design and implement. However, there are also numerous disadvantages to a uni-directional topology. The first disadvantage to the uni-directional topology is that the write command packets sent from the memory controller to the memory chips will be shorter than the read command data packets returned by the memory chips to the memory controller. This asymmetry in the length of the command and data packets means that the lengths of the packets must necessarily be different. The difference in the length of the command and write packets further implies that either the memory
chips must contain additional logic to dynamically delay and alter the flow and timing of the command and data packets, or idle cycles must be inserted into the access protocol to avoid conflicts on the data return channel. We illustrate a timing diagram for a packet based memory access protocol on an abstract memory system with uni-directional topology in figure 2. In figure 2, we show a command sequence of a read command followed by another read command, followed in turn by a write command, and the write command itself followed by a read command. In this sequence, we see that in step 1, a read command, shown as R0 in figure 2, is issued into the memory system. After the appropriate read request latency, a read data packet is placed onto the data return channel and returned to the memory controller. Then a second read command, shown as R1 in step 2, is issued into the memory system, and the data packet is returned by the memory system in response to the read command. We then show a write command packet, enclosed with the data associated with the write command, asserted onto the command channel in step 3, and the memory system is not required to provide a response packet to the write command. Finally, upon completion of the write packet, yet another read command, shown in figure 2 as R2 in step 4, is issued to the memory system.

Due to the CAS latency of the memory chips, shown as 7 cycles in figure 2, idle cycles occur on the data return channel after the R1 read command in figure 2. The appearance of idle cycles reduces the data transport efficiency of the memory system, and cancels out the efficiency gained by the elimination of the bus turnaround cycles. One alternative that allows for more efficient utilization of the command and data channels is the introduction of a dynamic timing system. In a dynamic scheduling system, the memory chips must contain additional logic to be able to accept read commands, then return data packets to the memory controller with variable latency. The flexibility in the return latency of a read command allows the memory controller to be able to

**Figure 2: Abstract Uni-Directional Bus Static Timing Access Protocol**
schedule memory requests with a higher degree of efficiency. In figure 3, we illustrate the same command sequence as shown in figure 2. However, we show that by allowing the data packet for the R1 read command to be delayed by a few cycles, the R1 read command may be issued into the memory system at an earlier time. Figure 3 shows that the data packets for read commands R0 and R2 are returned after the 7 cycle read latency, but the data for read command R1 is not returned until after 9 cycles of read latency. The end result here is that the dynamic scheduling allows the memory channel to be more efficiently utilized.

The combination of the command packet with write packet data sharing the same physical channel means that while a write data packet is being sent from the memory controller to a memory chip, a read command cannot be initiated to a different chip on the same channel. The consequence of this architecture means that read commands must be delayed by the memory controller until the write packet has been sent. In a heavily loaded memory system, the memory controller has many independent memory requests to schedule, and the sharing of the same physical channel will not impact the latency of a single memory request. However, in a lightly loaded memory system, where the average queuing delay is minimal, any delay in the issuance of a read request will negatively impact the average latency of the memory read requests.

In this study, we propose to compare and contrast the uni-directional topology against a bi-directional topology. In figure 4, we illustrate an evolutionary migration from RDRAM based
memory system topology. We attempt to modify the RDRAM memory system in order to retain the advantages and lessen the disadvantages of the RDRAM based memory system. In this evolutionary approach, we attempt to reduce memory request latency by increasing the number of pins used for the respective column and row command channels. The idea here is that by increasing the number of pins used in the command channels, the command packet could be transmitted in fewer cycles, and the memory chip could respond more rapidly to a memory request and result in reduced memory request latency.

We propose to examine the trade-offs between latency and bandwidth, using pin count as the constraint for the comparison of difference topologies. The examination of the topology will comprise one facet of our study in the efficiency of the proposed memory system. A second facet of our study will focus on the impact of channel length on the performance of the memory system. One interesting effect of a long channel with a large number of devices sharing the memory channel is that the long channel in general require longer memory read request latency as compared with the read request latency in a short memory channel. However, in order to keep the cost of the memory chips to a minimum, memory storage devices do not contain sophisticated logic, and can oftentimes only respond to a single memory request at a time. Successive memory requests to the same memory chip must then be queued until the current request is completed. The result is that with a longer channel, the chances of consecutive memory requests hitting the same chip is lower, and the memory controller can effectively pipeline the memory system by sending different memory requests to different memory chips in the system. The results is that a longer channel with more memory devices can attain a higher efficiency in the utilization of the memory system in a heavily loaded system. However, a shorter channel with few memory devices can then better respond with lower memory request latency to the infrequent memory requests in a lightly loaded system. Our study will seek to characterize the degree to which the trade-off between a long and short memory channel impacts the performance of a heavily loaded CMP or MT processor based system.