

Addressing interleaved multichannel memory challenges

INTERLEAVING ADDRESSES IN MULTIPLE DRAM CHANNELS CAN GREATLY IMPROVE MEMORY BANDWIDTH, BUT IT IS NOT A TRIVIAL TASK.

Achieving the total external-memory-bandwidth requirements of consumer SOCs (systems on chips) at acceptable costs gets more difficult with each generation of electronic systems and the DRAM technologies that serve them. SOC designers must optimize for total DRAM efficiency to enable the system to reach the highest performance and to minimize DRAM-subsystem costs. As SOCs move to more advanced DRAM technologies, such as DDR3, that have larger minimum burst lengths, challenges arise in delivering optimal memory throughput. Using multichannel DRAM subsystems helps maximize efficiency by ensuring that the DRAM bursts are smaller than the processor- and I/O-interface accesses.

WHY MULTICHANNEL?

Discussions of DRAMs introduce terms such as “banks,” “ranks,” and “channels” (Reference 1). Each term describes a mechanism for arranging multiple arrays of DRAM cells, either within the same chip or across multiple chips, to increase memory density, memory performance, or both. The multibank architecture of current SDRAM devices enables SOC DRAM controllers to improve performance by exploiting parallelism across the banks inside a DRAM chip to hide the page-closing and -opening penalties of the internal DRAM cell arrays. You could employ the same principle with multiple chips, but few

consumer SOCs support multiple ranks of DRAM devices, in which multiple DRAM chips connect to the same data signals to increase the total supported memory size, because few consumer-electronics systems need that much memory.

The substantial increases in performance in consumer SOCs require both more total on-chip processing and substantial increases in DRAM bandwidth. Although the data-pin bandwidth of DDR SDRAM devices has improved over time, these improvements have not kept pace with the requirements of several key consumer-SOC markets, such as HDTV (high-definition television). As a result, the total number of data pins necessary to satisfy DRAM-bandwidth needs is growing for such SOCs.

The combination of increasing minimum burst length associated with DDR3 and the increasing data-pin count leads to substantial increases in the minimum burst size of single-channel DRAM systems. When these bursts become larger than the data objects that the initiators on the SOC are accessing, effective throughput declines as the DRAM transfers become less efficient.

One approach to this challenge is to further increase the available DRAM bandwidth to compensate for the loss of efficiency. However, the bandwidth increase must come without further increase to the DRAM burst size, or you lose even more efficiency. A better approach is to introduce multiple channels.

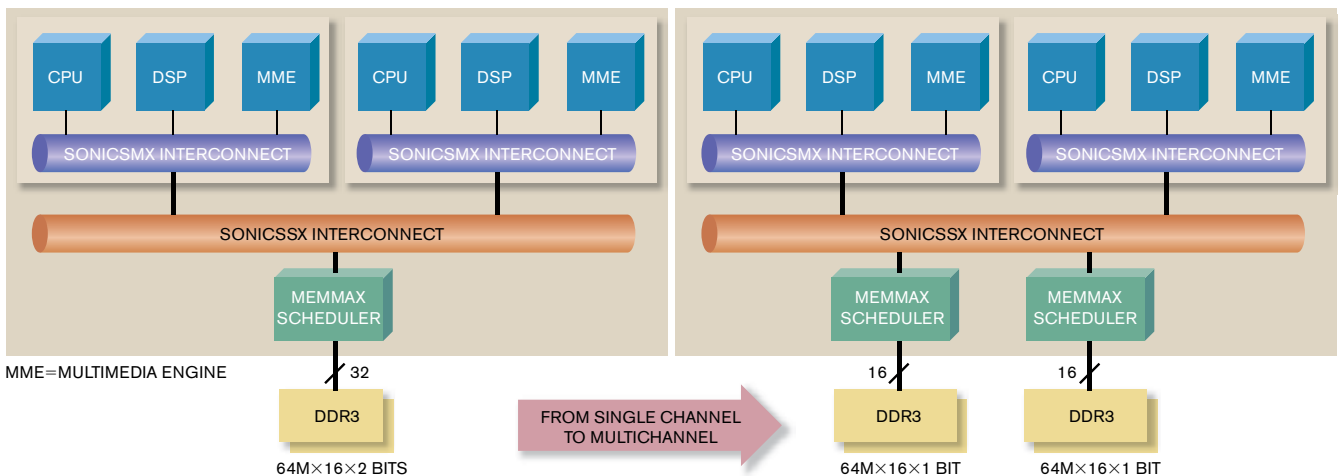


Figure 1 Moving from a single DRAM channel to two channels can increase peak bandwidth.

The key benefit of a multichannel DRAM system is an improvement in access efficiency due to shorter bursts that more closely match the size of the data types transferring to memory. Note that the DRAM bursts are smaller but not shorter because only the word is smaller. The prefetch degree of the SDRAM type in use still largely determines the burst length.

A second benefit of the multichannel system is a likely reduction in efficiency loss due to page-closing and -opening delays because an N-way multichannel system has N times as many banks and can therefore manage N times more open pages. However, the fact that the pages are only 1/Nth as large mitigates this benefit, so some transactions that might have found their page to be open in a single-channel system—due to a previous transaction’s accessing memory locations at a similar address, for example—can find a closed page instead in a multichannel system. In a consumer SOC with many initiators, enough data flows are often waiting for access to DRAM that the benefit of more open pages outweighs the smaller page.

To quantify the efficiency benefits available from a multichannel memory system, consider the migration of a production HDTV SOC from a DDR2 SDRAM baseline to a next-generation design based on DDR3 devices. A static analysis of the memory traffic from the original design—currently in high-volume production—assigns the memory efficiency to be 100% as a point of reference. For the next generation of this SOC, compare two memory-system architectures: a single-channel DDR3 system with a word size of 32 bits and a two-channel system in which each channel has a word size of 16 bits (Figure 1). Note that both configurations offer the same peak DRAM bandwidth and use the same number and type of DRAMs.

You might expect the single-channel DDR3 design to have lower memory efficiency because DDR3 devices use a prefetch-8 architecture and thus have a minimum efficient burst length of 8 words, whereas the reference system has the 4-word burst characteristics of DDR2. When you consider the efficiency loss due to both minimum DRAM bursts that are larger than the SOC traffic patterns and address-alignment problems that also result from larger bursts, you can see that the single-channel DDR3 system loses a substantial amount of efficiency. In contrast, the dual-channel system has longer bursts that are half as wide, so the basic burst size and alignment do not change. Therefore, the dual-channel DDR3 system is as efficient as the single-channel DDR2 system of the same size. Table 1 shows the efficiency difference, which directly translates into usable memory-system bandwidth.

Note that this static analysis ignores any of the bandwidth increases that are available due to the higher operating frequencies available with DDR3 SDRAMs. It compares only the relative efficiency of the memory usage. It also ignores the likelihood that the DDR3 design will have higher overall performance requirements and, thus, higher total external-memory-bandwidth requirements, which are likely to outstrip

THE KEY BENEFIT OF A MULTICHANNEL DRAM SYSTEM IS AN IMPROVEMENT IN ACCESS EFFICIENCY.

the frequency benefits of the DDR3 transition. More important, the analysis assumes that you can schedule the memory-system traffic for maximum efficiency in all cases and balance it evenly across the dual-channel configuration.

Most scenarios require you to balance traffic across the channels to deliver multichannel benefits. For example, applications that require high memory efficiency

normally have a fair amount of queuing to cover the latency between memory requests and responses, including the arbitration delays on the SOC among the initiators. If a lightly loaded channel runs out of requests to service while the other channels are busy, then that channel’s throughput and the efficiency benefits of the multichannel approach decrease. If the channels act as independent regions in the memory map, the SOC architect, accelerator designer, firmware developer, or application developer should carefully allocate data structures in memory so that the initiators’ access to those structures is well-balanced over periods as short as a few microseconds. It is difficult to manage this task with a dual-channel memory system and nearly impossible with the four-channel systems of the future.

Another challenge with this static load-balancing approach is that transaction-ordering requirements often prevent a single initiator from sharing memory bandwidth from multiple channels at once. This problem arises from the nature of the initiators’ communication protocols, which specify that request and response order should match, and from the fact that DRAM channels have significant latency variations. So the response to a first request to a first channel may likely be unread for delivery until after the response from a second channel is ready.

Using flow control to hold off the second channel is typically unacceptable because it would cause the DRAM to stop servicing requests while waiting, which reduces DRAM throughput. The result is that static load balancing generally requires that most initiators

communicate only with a single channel, which makes sharing and load balancing substantially more difficult.

WHY IS INTERLEAVING THE ANSWER?

The ideal approach for load balancing a multichannel DRAM system would be one that achieves excellent balancing of traffic, is largely independent of the number of channels, and requires no extra work in the design of either the initiators or the software that controls them. Rather than treating the channels as independent memory regions with the resulting load-balancing challenges, interleaving the channels in the address space enables them to appear as a single, logical memory region and offers the promise of achieving all of these goals.

To understand why channel interleaving can achieve automatic load balancing, it is important to understand the memory-access behavior of the initiators in the SOC. The initiators access data structures in DRAM, so the type of stored data or the processing algorithm in use with the data determines the expected access patterns to those data structures. Streaming accelerators, communication processors, cameras, displays, and

TABLE 1 MEMORY-SYSTEM EFFICIENCY

| | DDR2 | DDR3 | DDR3 |
|-------------------------|------|------|------|
| Channels | One | One | Two |
| Data word width (bits) | 32 | 32 | 16 |
| Effective bandwidth (%) | 100 | 84 | 100 |

CHOOSING THE INTERLEAVING BOUNDARIES IS A KEY TASK IN ACHIEVING GOOD LOAD BALANCE.

I/O interfaces all normally manage large data buffers in DRAM. They typically access those data structures using mid-sized to long burst transactions, with each burst transaction starting at the sequential address following that of the previous transaction, until they reach the end of their buffer. In contrast, a CPU accesses a variety of data structures in memory, from small control structures through large buffers. Because most CPUs in consumer SOCs have internal caches, the dominant accesses to DRAM are cache-line-sized, and the principles of spatial and temporal locality teach that many of the accesses in a time window have similar addresses.

For many consumer SOCs, the video decoder offers a particularly challenging case. The video decoder accesses several frames of uncompressed video images, each of which may comprise 2 million pixels of storage, and an incoming stream of compressed data that tells the decoder how to decode the next frame. The incoming stream frequently tells the decoder to fetch an arbitrary macroblock from the frame storage, with each macroblock comprising a 2-D set of pixels. Because the required macroblock can start at any pixel address, it is unlikely that the macroblock will fit nicely into a few DRAM bursts. Furthermore, the total required macroblock bandwidth is high enough that mapping the macroblocks into efficient DRAM transfers is essential.

Earlier interleaving often focused on improving memory bandwidth by accessing multiple physical DRAMs in an interleaved manner to improve pin bandwidth. These systems relied on spreading a burst transaction across several DRAMs. This approach differs greatly from one in which you intentionally create smaller channels of memory to reduce, rather than increase, the burst size. Some of the latest PCs, including servers and desktops, also use multiple interleaved channels. These channels are normally interleaved at CPU cache-line boundaries in an attempt to best exploit the spatial locality of computing applications. These boundaries are finer than optimal for consumer SOCs, in which the access patterns tend to have more regularity.

The choice of interleaving boundaries can greatly affect the load balancing. Some of the accelerators access data structures in a fairly predictable pattern, in which the address of the next memory request is spaced a fixed distance away from the previous one. These strided accesses reduce the channel balance when the stride value is a multiple of the interleaving size because consecutive accesses may map to the same channel. Because there can be a close correlation between memory data structures and strided access patterns, the designer can optimize overall throughput by selecting different interleaving boundaries for different regions of memory based on the data structures they each store.

The challenge of balancing the traffic loading increases with the number of channels. The interleaving approach can balance the traffic as long as most initiators regularly access each of the channels—in other words, as long as the number of channels times the size of the interleaving boundary is smaller than the range of addresses the initiators are accessing. Thus, optimizing the interleaving boundaries is related to the number of channels. Choosing the interleaving boundaries is a key task in achieving good load balance, and you may achieve better balance when you divide the DRAM address space into several subregions with different interleaving boundaries. Additionally, because the best boundary choice can depend on the data structure and access patterns and because these parameters can both change based on the operating mode of the SOC, it is sometimes valuable to change these boundaries when the mode changes.

A major benefit of interleaving the channels is in isolating the initiators from the channel configuration, which enables simpler design and much greater reuse of the processors, accelerators, and software for the SOC. However, this isolation ensures that the initiators are not channel-aware, thus increasing the importance of maintaining high throughput for initiators that are accessing several channels at once. Computer systems maintain throughput by building reordering buffers near the initiators,

so channels that service requests sooner than an initiator is ready to receive them must have a place to store their responses. However, the large number of initiators in a consumer SOC and the growing depth of the DRAM-access pipeline mean that the total amount of required storage for this reorder buffering would be too large and expensive in the markets these SOCs serve.

It is equally important to ensure that the SOC memory controllers have a great degree of flexibility for scheduling traffic to the DRAM channels to ensure the highest efficiency and throughput. The interleaving system should therefore limit neither the number of transactions that can be outstanding to channels nor the controller's ability to schedule the transactions that it has received.

One approach that could address many of these challenges would be to manage the interleaving in the memory controller itself. This approach has the advantages that it localizes the information about the number of channels and in-

INTERLEAVED MULTI-CHANNEL TECHNOLOGY MANAGES FLEXIBLE INTERLEAVING BOUNDARIES AMONG MULTIPLE DRAM CHANNELS IN THE INTERCONNECT.

terleaving boundaries into one location on the SOC, minimizing the disruption to other architectures and hardware, and allows the use of shared buffering in the controller to manage ordering and allow many transactions to be outstanding across the channels. However, such an architecture requires most of the system communications to pass through a single point on the SOC, which is likely to create a performance bottleneck in both wire routing and memory efficiency. This memory-efficiency loss can result from the same access-granularity problem that happens in a single-channel DRAM system: that the internal interface carrying the memory traffic may become so wide—to carry the total DRAM bandwidth for the SOC—that a DRAM burst for a single channel may not pack effi-

ciently into the internal-interface word.

The SonicsSX interconnect from Sonics (www.sonicsinc.com) uses another approach, which employs IMT (Interleaved Multichannel Technology). IMT manages flexible interleaving boundaries among multiple DRAM channels in the interconnect, rather than in the RAM controller, providing the benefits of automatic load balancing and high throughput without creating performance bottlenecks or requiring reordering buffers. You measure the DRAM efficiency as the fraction of the DRAM clock cycles during which a useful data word is transferring to or from DRAM. Although achieving DRAM efficiency of 60% is relatively straightforward for most designs, targeting efficiencies of 75 to 90% is more challenging and normally requires substantial analysis and optimization during the SOC-design phases.

Choosing the right multichannel architecture involves selecting the proper number of DRAM channels, allocating the DRAM address space across one or more multichannel memory regions, and selecting the interleaving characteristics for each region. The SOC designer normally chooses the minimum number of channels that provide the required memory-system efficiency and throughput. This configuration generally minimizes system costs because it results in the minimum DRAM costs and reduces the number of DRAM-related control and address pins on the SOC.

Single-channel SOCs normally treat DRAMs as single pools of address space that all of the initiators share. The software that executes on the host CPU during booting allocates some of this DRAM to specific uses and initiators, and the operating system dynamically allocates the rest of the DRAM space. In an interleaved multichannel system, the strided access patterns of some initiators can cause channel imbalances with certain interleaving boundaries. When several such initiators share the memory system, the designer may wish to allocate multiple subregions in the logical DRAM address space with different interleaving boundaries that better match the access characteristics of the initiators that share each subregion.

When multiple subregions are in use, the operating system normally allocates one, and designers have optimized this

subregion for more general-purpose traffic, typically with a small interleaving boundary—perhaps at the cache-line level, as in a PC. The booting process or the device drivers normally assign the other subregions for collections of initiators with more predictable ac-

cesses. Such subregions are likely to have larger interleaving boundaries. Designers can support multiple operating modes with multiple address subregions by allocating more total DRAM address space than the DRAMs contain and allowing the subregions to alias their contents on to each other in memory. This approach relies on careful allocation of data structures in the subregions to ensure that you do not simultaneously allocate a given area in physical DRAM to multiple data structures in different subregions.

Designers normally map each subregion onto all of the physical DRAM channels. But aggressive power-management schemes, in which some of the channels may power down in some operating modes, provide an example in which a designer may populate and power some subregions with only enough channels to deliver the required memory throughput for those modes.

Once designers know the number of channels in a subregion and the initiator-access characteristics, they can choose the interleaving boundary for the subregion. The choice of an interleaving boundary is critical for achieving good load balance among the traffic that targets that subregion.

ESTIMATING RESULTS

SOC designers do performance estimation and analysis using spreadsheets, cycle-accurate simulation in SystemC, and RTL (register-transfer-level) simulation. Although each technique has its advantages and disadvantages, many designers apply several of the techniques to a design. It is therefore important to have a consistent vocabulary for performance-oriented characteristics of the design with instrumentation to measure or calculate those characteristics. Key measurement parameters include throughputs, latencies, and DRAM channel efficiencies, which you measure in the timing domain of the initiators that gener-

✚ Go to www.edn.com/ms4326 and click on Feedback Loop to post a comment on this article.

✚ For more technical articles, go to www.edn.com/features.

ated the traffic. It is useful to have access to these performance results both as aggregated summary data and at the granularity of individual transactions.

SOC designers must compare these performance results with the quality-of-service requirements of the

system. Knowing the granularity of individual-transaction results helps SOC designers debug simulation results to understand why the performance may be different from what they expect. Tooling that helps designers track transactions as they propagate from the initiator, across the interconnect, through the memory scheduler, and into the DRAMs increases visibility into challenging performance-debugging situations. Once designers gain insight into the reasons for performance results, they can use it to modify the SOC and memory-system configuration to further optimize performance.

The substantial increases in DRAM-bandwidth requirements of consumer SOCs and the prefetch architecture of DDR SDRAMs have caused single-channel DRAM systems to lose substantial efficiency as bursts become larger than SOC transactions. This scenario leads designers to select multichannel DRAM systems. However, only interleaved multichannel systems that support multiple subregions with different interleaving boundaries deliver the automatic load balancing and hardware and software transparency necessary for consumer SOCs. **EDN**

REFERENCE

1 Wingard, Drew, "DRAM technology for SOC designers and—maybe—their customers," *EDN*, Aug 6, 2009, pg 34, www.edn.com/article/CA6674038.

AUTHOR'S BIOGRAPHY



Drew E Wingard, PhD, co-founded Sonics in September 1996 and is currently chief technical officer and secretary. He received a bachelor's degree in electrical engineering from the University of Texas—Austin and master's and doctorate degrees in electrical engineering from Stanford University (Stanford, CA).