O2™

SiliconGraphics Computer Systems

UNIFIED MEMORY ARCHITECTURE

Summary

The O2™ system Unified Memory Architecture (UMA) represents a radical departure from that of earlier workstations or personal computers. The traditional workstation uses multiple high-speed buses to connect numerous subsystems and their individual local buffers (e.g., frame buffer, texture, z-buffer, or image memories). The O2 system instead uses ultra high-speed multiported main memory to allow all data types to be stored in one memory array.

The benefits of the O2 UMA are:

- Lower overall system cost
- More efficient use of the entire system memory
- More flexible operation because all data is accessible to the CPU
- Increased performance due to reduced buffer copies
- Lower cost for memory upgrade of subsystem memories

These features of O2 UMA make this possible:

- The O2 UMA removes the need to copy data from one subsystem to another, thereby eliminating typical system bottlenecks (e.g., texture upload time).
- Specialized local memory buffers have been eliminated from each subsystem, significantly reducing overall system cost.
- Efficient use of memory resources because unused memory is returned to the general CPU memory pool for use by other subsystems or applications.
- Expensive multiple high-speed buses have been replaced with a single ultra high-speed connection to memory, significantly reducing total system cost.
- Memory upgrades (e.g., texture memory) involve adding standard commodity synchronous DRAM (SDRAM) since subsystem buffers simply consist of CPU memory.
- UMA allows processors to specialize by type of computation, not by algorithm or application.

The key to an effective Unified Memory Architecture is ensuring that each subsystem has enough memory bandwidth so that it is never starved for data. The O2 system has more than 2.1GB per second of overall memory bandwidth. All DMA subsystems can perform simultaneously at full speed, without hitting a system bottleneck.

Fundamentals

A computer system architect must consider three factors: cost, features and performance. A computer system design can achieve a range of different performance levels, but generally the cost of a performance level rises exponentially with increasing performance. The computer can include a wide range of features, however, these costs generally rise linearly with the number and complexity of the features.

The architect must also know at what level innovation is possible. In UNIX® systems, for example, an architect has full control over the hardware and software and can innovate at any level. In the Windows® BIOS environment, however, certain areas of the system, notably the CPU/memory subsystem, are not available at an architectural level for innovation.

Because system architectures are defined one to three years in advance, an informed understanding of what technology is expected to be commercially available in that time frame is key to designing a workstation with the best performance at the lowest cost. For example, when the O2 system was being designed, it was clear that synchronous dynamic random access memories (SDRAM) would be available at commodity prices in the O2 introduction timeline. Commodity pricing of SDRAM allowed the O2 system architects to rethink the differences between traditional bus-based and the O2 unified memory architectures.

Traditional systems are forced to continue supporting certain legacy features. The Windows BIOS computer model, for example, restricts the user from accessing the high-speed memory bus that connects the CPU and main memory in order to maintain compatibility with PCI/ISA devices and associated software. A Windows system architect must work within other such constraints to maintain compatibility with Intel® and Microsoft® standards.

For O2, the primary legacy requirement is binary application compatibility with the rest of the Silicon Graphics® product family. This legacy requirement imposes essentially no architectural limitations at the system level because the majority of the compatibility is provided at the CPU level.

In summary, a system architecture is a complex juggling of economics, futures, legacy requirements and limitations established by a business model. The O2 architecture is able to provide industry-leading features, benefits, and price/performance in its product class because of innovation at every level—from the base system to the subsystems.

O2 Architecture

Overview

The O2 system is based on a Unified Memory Architecture (UMA) and an extremely high bandwidth, 2.1GB per second synchronous DRAM-based main memory. This main memory contains all data: video, image, texture, compression, and application. The main memory array is multiported with a memory bandwidth budget sufficient to ensure that no subsystem is ever starved for data.

UMA allows the processing subsystems to share all data seamlessly and efficiently. For example, uncompressed video data is read and stored in main memory. This data can then be used as a texture by the 3D graphics subsystem merely by passing a pointer; no data copy is necessary. The ability to share data without buffer copies, especially across an external bus, removes a major bottleneck in digital media and 3D manipulation.

UMA also reduces the total amount of memory a system needs to perform a wide range of operations. In a typical PC system, for example, each subsystem has its own local memory. The video subsystem has its own memory, as does the texture map subsystem, the image processing subsystem, and so on. PC local memory can only be used by the specific subsystem to which it is attached. When the subsystem is inactive, this memory is unused. In the UMA-based O2 system, unused memory is returned for use by any other part of the system, including applications.

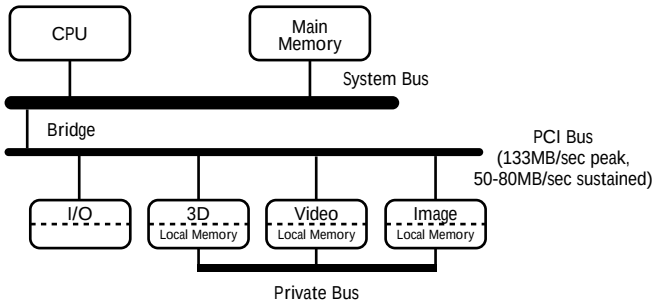
Unified Memory Architecture

A Unified Memory Architecture is one in which all buffers in a system are combined into one common pool, the system's main CPU memory. Each subsystem, such as the image processing or the graphics subsystem, has full access to not only its own data, but also to data from each of the other subsystems without a buffer copy.

The best way to understand a Unified Memory Architecture is to compare it to the PC architecture.

Typical PC Architecture

In contrast, a non-unified memory architecture creates islands of data connected by buses to each other or to main memory. This type of architecture is typical of the personal computer as illustrated in the figure below.



In this typical case, the Microsoft/Intel Windows BIOS architecture specifies that only the CPU and the main memory subsystem can reside on the system bus. All other subsystems must reside on PCI or ISA and communicate with the CPU and main system memory through a bridge.

Each subsystem must share a PCI bus capable of handling between 50 and 80MB per second sustained. Since modern UltraSCSI channels typically consume 30MB per second sustained per channel, the PCI bus is quickly saturated.

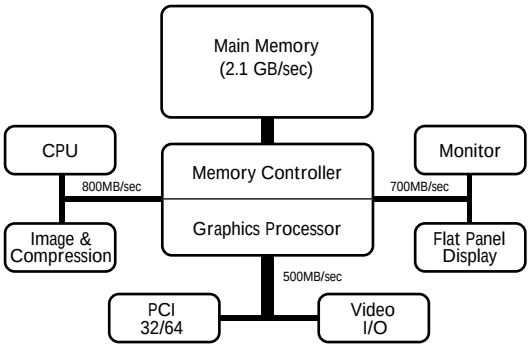
Designers of specialty boards typically resort to connecting the individual subsystems together with a private high speed bus. This bus allows controlled, real-time transfers among the local buffers on each board.

In this architecture, the CPU has no real-time access to the data in the local buffers. The moment that this data is placed on the bridge to the system bus, the system bottlenecks and the data access is no longer in real time.

In addition, the specialty boards typically run an entirely different operating environment than Windows or Windows NT. A real-time kernel is needed to manage data movement and processing. The Windows operating system on the main CPU does little more than run the GUI and pass control information to the subsystems. The Unified Memory Architecture of the O2 system is designed to prevent this bottleneck and add flexibility.

O2 Unified Memory Architecture

The O2 Unified Memory Architecture is based around a four-bank, multiplexed SDRAM subsystem that delivers a sustained 2.1GB per second of bandwidth. Each subsystem accesses this memory through high-speed ports designed to ensure that no data starvation occurs on any port.



Local Buffers

All of the subsystems in the O2 use the main memory for buffering data, which significantly reduces the total cost of the system. There is no need in the O2 UMA for separate buffers for video, graphics, image processing, or compression. The data types which are stored in main memory include:

- Frame buffer for up to 32-bit double buffers, z-buffer, stencil planes, and texture maps
- 2D image data
- Uncompressed YCrCb video data in CCIR601 non-square or square pixel format
- Application data and the operating system kernel
- pbuffers for nondisplayed frame buffer rendering
- JPEG, MPEG-1, and H.261 data for decompression

Reduced Data Movement

The O2 UMA dramatically reduces the amount of data movement in the system. In a typical bus-based system, data sent from one subsystem to another has to be transferred directly via a private bus; or if no peer-to-peer connection has been established, via a copy first to memory and then to the receiving subsystem.

Since all data types are kept in main memory, digital media data can be transferred from a data source to a data sink by simply passing a pointer. Since the data sink uses the same buffer as the source, no data movement is needed. This technique removes milliseconds to seconds of transfer latency from these operations.

In order to simplify this new method for driver level programs, the O2 UMA supports a new software construct called digital media streams (dmstreams). The dmstreams structure is similar to UNIX streams. When passing data from one subsystem to another (e.g., from the rendered pbuffer to the texture mapping engine), a dmstream connection is established between the source and sink. If the data requires no conversion, only a pointer is passed to the data sink. If a conversion is required, a conversion module is placed automatically between the source and sink.

Efficient Memory Usage

The O2 UMA not only lowers the cost of the system by reducing the number of local buffers; it also dramatically increases the efficiency of memory use. Memory is only allocated when needed. When it isn't needed, it is deallocated and made available to other subsystems or applications.

One example is the size, depth and complexity of the frame buffer, z-buffer, and texture memory. A demanding visual simulation application might require a 32-bit double-buffered frame buffer with 24-bit z-buffers and 64MB or more of texture memory. Allocating all of these features can require over 70MB of memory. In a bus-based system, the user would have to buy more than 70MB of dedicated graphics memory, typically at prices significantly higher than normal commodity DRAM prices. With O2, if the overall system has sufficient CPU memory, then the application will run. If the user needs to buy more memory, the upgrades are available at typical commodity memory prices. And with O2, when the application finishes its execution cycle, the memory that was used for textures, deep frame buffers, and z-buffering is now available for use by other subsystems or applications.

CPU Viewable Data

A key advantage of the O2 UMA is that the CPU can view and operate on all data in real time. In a typical bus-based system, moving data to main memory for the CPU to operate on means that the data can no longer be used in real time by the subsystem. The CPU can only operate on "snapshots" of digital media data.

Although subsystems are increasingly intelligent, there will continue to be a wide range of infrequently performed functions that don't justify dedication of gates on an ASIC. An alternative solution chosen by a number of subsystem manufacturers is to place another CPU on the subsystem module to operate on the local data, further increasing the cost and decreasing the flexibility of the total solution.

The O2 CPU can view all data types in the system. This capability means that an application can support a wide range of features efficiently and in real time by executing the actual data available to the subsystem.

Price/Performance of Additional Features

The subsystems included in the O2 system (the 3D graphics, CPU, compressed and uncompressed video, and image processing) are the sets needed to flexibly and seamlessly manage digital media and 3D data together. The O2 UMA means that large incremental costs weren't added with each subsystem.

Applications that mix digital media and 3D data provide capabilities that were previously not possible at the O2 price point, such as real-time video texturing. For example, in military training, it is desirable to import video data in real time from an infrared sensor. The data must then be image processed to either reduce the noise, or in some training situations, to increase or change the characteristics of the noise. This processed video image is then used as a texture in a real-time 3D scene. The entire training session is captured as it happens in a JPEG compressed movie file for later review by a collaborator.

The O2 UMA allowed these dedicated subsystems to be added with low incremental cost. The O2 avoids the typical bus-based system requirement of adding local memory and dedicated buses when adding subsystems.

In the O2, the system was designed to have enough memory bandwidth available to support each function; the cost of adding a feature then became that of the engine itself and a single port to main memory.

Conclusion

The O2 workstation is the result of architectural innovation at every level of a computer system. Silicon Graphics engineers were able to merge CPU and main memory subsystems with 3D graphics and digital media engines in a way only available to a company that controls the entire architecture.

The Unified Memory Architecture enables Silicon Graphics to offer a combination of price, performance, and features which is unparalleled in the low-cost UNIX workstation and PC/NT marketplace.

1352 (4/97)