White Paper

An Explanation of the Advantages Afforded by QED RM5200 and MIPS R12000 CPUs in O2

Authored by: Ian William

The Silicon Graphics O2 was a catalyst to broad acceptance and adoption of integrated graphics and digital media in the low-end UNIX desktop market. Through its highly innovative architecture, it revolutionized the way graphics features including texture mapping and digital media technologies, such as video capture and playback, could be delivered in an affordable system.

The O2 architecture allows these diverse types of data to be created, manipulated, and integrated readily, allowing users to work much closer to their mental vision of their products. Those products can be as diverse as an oil drilling platform or an animated cartoon character. Indeed, within the UNIX market many O2 capabilities remain unsurpassed, even when compared to significantly more expensive systems.

The upgrade of the O2's CPUs from the MIPS R10000 to the MIPS R12000 and the MIPS R5000 to the QED RM5200 allows the platform to deliver significant performance increases in both computational and graphics-related usage. This white paper describes the benefits the new CPUs bring to both low-level graphics and real-world application performance in markets where O2 adds real value. It also demonstrates O2's strengths relative to competitive systems in these markets. Compared to many O2s used by existing customers, the performance of systems incorporating the new CPUs can provide end-user productivity gains significantly more than 100%.

White Paper

Table of Contents

Introduction – Current State of Play

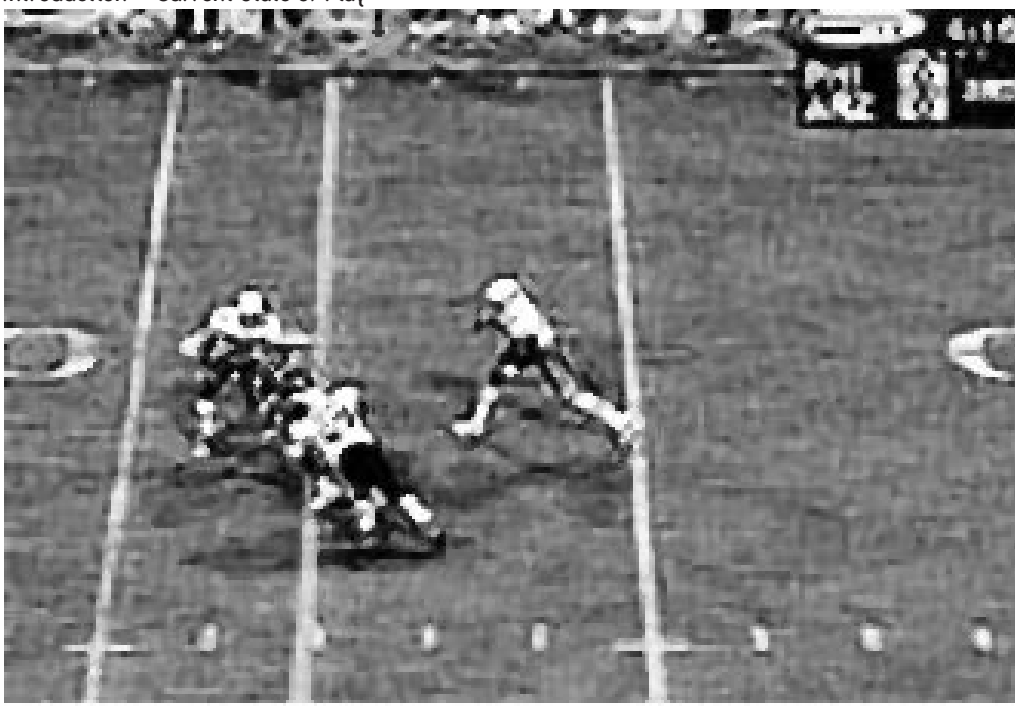


Figure 1, "The '1st and 10' line makes its mark" [1], [2] and Appendix I.

Most readers will at least recognize the image in Figure 1, even if they aren't avid American football fans. At first glance though, it would be easy to miss its relevance to this paper and what it really represents. On closer study, those readers more familiar with American football may speculate on the state of play or even possibly the outcome of the game. Most non-football fans probably wouldn't pay much attention to the yellow line marking the 1st down line. Although a simple element of the picture, the yellow line represents a significant step forward in technology made possible by the architecture and integrated features of the Silicon Graphics O2 system. Its oversight by many readers should, therefore, be taken more as a compliment than an insult.

A few years ago it wasn't practical to contemplate using a simple visual effect such as superimposing a yellow line onto live video footage. Now, though, it is a fairly familiar sight, especially to American football fans. The Silicon Graphics O2 system made this possible by integrating graphics and digital media technology in a seamless and efficient way at a low cost. It affords many unique advantages compared with competitive systems in its class and offers features that were previously only available on dedicated one-off systems that were significantly more expensive, in some cases even up to orders of magnitude so. As a result of the O2 system, it's realistic to contemplate using these features on a day-to-day basis, which clearly demonstrates significant real-world benefits to end-users.

Because the example presented relates to the broadcast industry it would be easy to think that the benefits afforded by O2 are restricted to this and similar markets. The distribution of O2 units across different market segments, shown in Figure 2, clearly demonstrates O2's widespread appeal across all industry segments.

Figure 2
Breakdown of O2 Markets

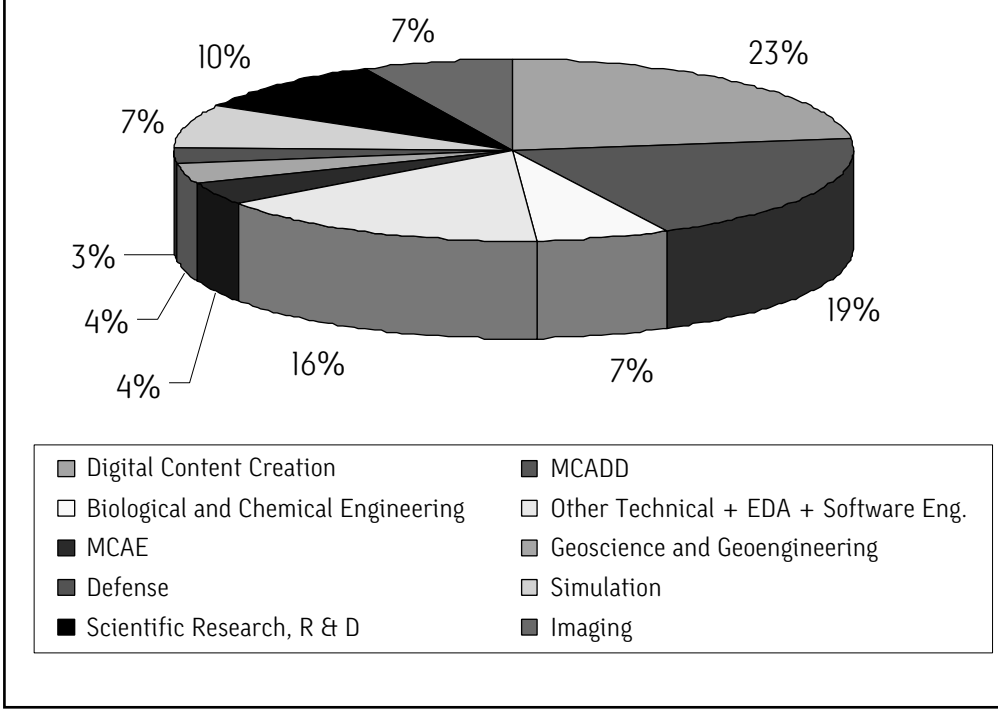


Figure 2, Breakdown of O2 market segments based on units/revenue.

Clearly, the greatest number of O2 systems are sold into the digital content and creation (DCC) market, which includes the application presented above. This is because in this market both O2's architecture and features make it an obvious fit. The Mechanical Computer Aided Design and Development (MCADD) market is a close second. In this case, features such as client scalability coupled with a balance between visualization and compute capability make O2 an extremely attractive low-cost proposition in the UNIX marketplace. Likewise, these features also make O2 attractive to the Scientific Research and R&D, and Simulation, Imaging, and Biological and Medical Engineering markets.

Given O2's clear appeal across all these markets, what aspects of its design contribute to its popularity? The next section aims to address this question by describing in detail many of the unique features of O2's architecture, highlighting differences to competitive systems and architectures, and showing how these differences translate into end-user benefits.

The End-User Benefits and Competitive Advantages Afforded by O2's Architecture

Figure 3, Diagram of O2 system architecture.

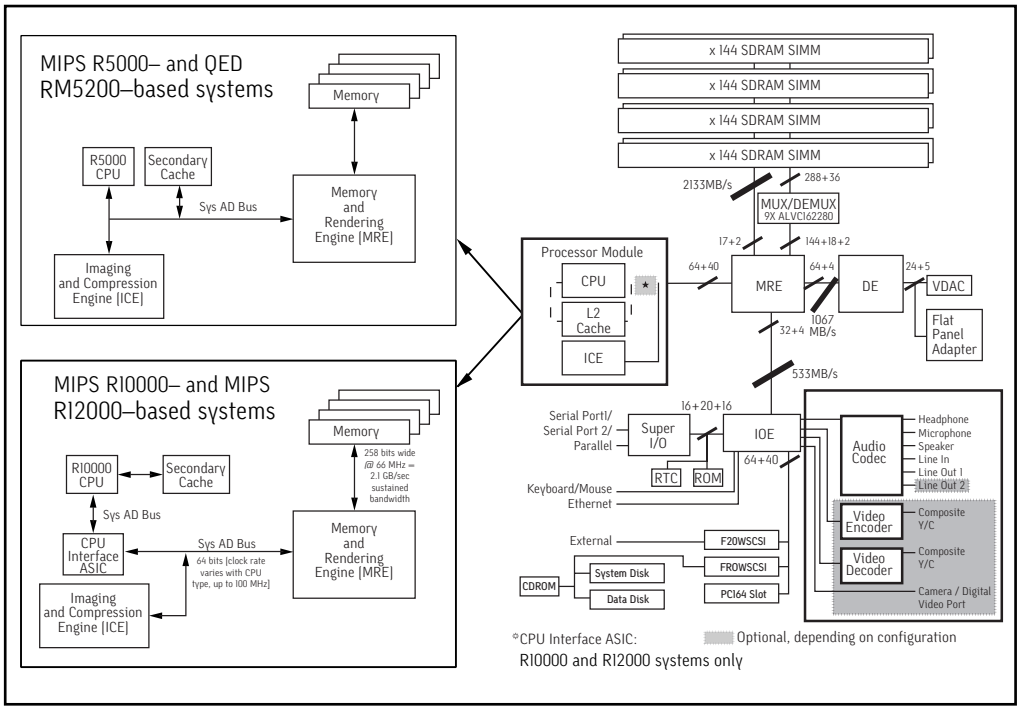
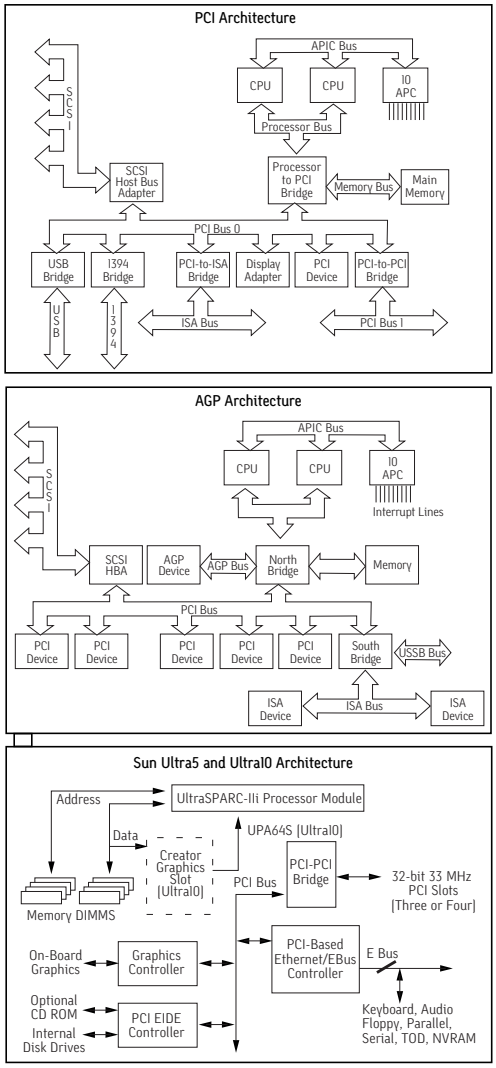


Figure 3 shows a detailed block diagram of O2's architecture. As can be seen from the diagram, it comprises five key units: main memory; the Processor Module, which includes secondary cache and the Image Compression Engine; the Memory Rendering Engine; and both the Display and I/O Engines.

Many readers will be familiar with the function and operation of each of these units and, to a high level at least, they should be fairly self-explanatory. Some of the subsections below, such as those discussing the technical strengths of O2's architecture in more detail, as well as later sections in the paper, assume more than a very high-level appreciation of O2's architecture. In such sections sufficient background information is given to appreciate the points being discussed; however, inevitably some details probably don't receive as much attention as perhaps they should. To this end readers are invited to refer to [3] for further details and information on O2's architecture and components.

For comparison purposes, Figures 4a and 4b show generic low-cost architectures based on PCI and PCI combined AGP, respectively. Figure 4c shows the architecture of Sun's Ultra5 and Ultra10 systems. For more information and details on both PCI and AGP architectures and capabilities, see [4] and [5]. For more information on Sun's Ultra 5 and Ultra10 system architectures, see [6], and see [7] for further information on Sun's UPA graphics connection.



Figures 4a, 4b, and 4c. Diagrams of generic PCI bus-based architecture; generic architecture using both PCI and AGP busses, and Sun Ultra5 and Ultra10 architecture.

Clearly, when comparing O2's architecture to typical competitive systems it is easy to see that it represents a major departure from the usual solutions and hence it may be tempting to think of it as unnecessarily complex. Apart from perhaps being somewhat short-sighted, this misses several key points. To explain the motivating factors behind it, consider an analogy of traveling by car or airplane. A journey by car at first may appear significantly less expensive than a journey by aircraft, as well as offering much greater convenience. These advantages, however, reduce dramatically when traveling more than a few hundred miles. Taking into account the true cost of ownership, as well as cost implications arising from time spent traveling, it is easy to see why the car soon becomes significantly less attractive when covering more ground. Also, a major factor in the apparent convenience afforded by a car arises from the ability to travel between two destinations via several alternative routes, as occurs within suburbs and cities. Over greater distances these routes typically amalgamate into a small number of key routes, between states or countries, say, and hence offer no advantage over the defined routes used by aircraft.

As the example clearly illustrates, when traveling between any two destinations most people choose between a car or an airplane depending on which is more appropriate for the length of the journey as well as other factors. The reason for O2's departure from more traditional architectures was, therefore, to produce a system that allowed users to translate and manipulate their mental vision of an idea in an easy, effective, and efficient way. To do this requires being able to manipulate and combine diverse types of digital data, such as 3D geometry, images, video, or audio, in a seamless and integrated fashion. In the same way, people in the real world aren't restricted to always using a car or an airplane for all journeys regardless of length. O2's architecture allows the most appropriate data to be used to describe some aspect of an idea or product and since the capability is offered at a low-cost, it exemplifies why O2's architecture can be considered groundbreaking and also represents a major step forward in the computing industry.

At first sight it may appear as though O2's architecture would result in a more expensive system compared to utilizing CPU performance to perform many activities. It should be appreciated, however, that once a design is committed to silicon, production costs are very low. It's interesting to note that the cost of additional CPUs in competitive machines is significant when compared with the price of an O2 system. Of course, due to its unique architecture, O2 doesn't require additional CPUs for many sophisticated tasks. The benefit of O2's architecture to the end-user is clearly that it not only delivers both advanced and sophisticated digital media capabilities at a very low cost, but it also doesn't incur significant additional costs, such as multiple CPUs, to be able to fulfill expectations and deliver acceptable performance.

In many situations it is appropriate to use system resources for a variety of uses, for example, when using main memory instead of dedicated texture or video memory. The design of O2 incurs no performance penalty and affords considerable flexibility to meet widely varying needs of applications across all marketplaces. It also reinforces the strengths of O2's design in delivering sophisticated features and functionality as well as being able to cope with many conflicting requirements for low system cost. After all, with the exception of Silicon Graphics' 320 and 540 platforms, no other technical workstation currently sold under $8K provides up to 1 MB of texture memory and offers as many sophisticated digital media features, such as real-time uncompressed video capture/playback and real-time 3D video textures.

High-level intent and good design, however, do not directly equate to end-user benefits. To demonstrate the benefits afforded to the end-user by O2, it's necessary to consider specific examples of digital media technologies and how they relate to typical user activities. In this way the strengths and unique features of O2's architecture can be highlighted in comparison to alternative system architectures and the end-user benefits demonstrated. To this end, the remaining part of this section concentrates on three graphics and digital media technologies: texture mapping, video processing, and imaging, showing how they relate to typical user activities and the benefits afforded through O2's architecture.

Texture Mapping

Most readers are probably very familiar with the concept of texture mapping. Indeed, in the Digital Content Creation and Architectural Engineering marketplaces it is an accepted feature in widespread use. A good description of the many incarnations of texture mapping is presented in [26]; however, perhaps the most widespread use is where it appears as if a two-dimensional image is applied to a 3D object within a model or scene. This dramatically increases the realism of computer or digital representation. Figure 5 shows an example taken from an application called Review Reality produced by CADCENTRE (UK). Texture mapping can also be used to create better lighting effects on models and components, such as would be needed to represent correctly the effect seen when shining a particular spotlight on an object.

Figure 5, An example of texture mapping taken from the CADCENTRE Review Reality program.

To assist a user's creativity and be of real benefit, it is important to be able to interact with such a model or scene so that the full implications of individual components can be appreciated. When using texture mapping, doing this requires the image data to be referenced each time the scene is drawn. If manipulation is to be interactive, this equates to many times a second. To illustrate the system requirements this can create, consider that a true color image 1280x1024 pixels in size corresponds to 3.9 MB of data. Even though 1280x1024 pixels may be considered large relative to some images, it does correspond to the display resolution of most workstations as well as the current resolution of higher-performance digital cameras. Likewise, drawing a model or scene 60 times a second may be considered fairly optimistic; however, it is considered a good threshold above which the eye is unable to discern discrete movement, and hence most good computer displays refresh at this rate and above.

Transferring the image between main memory and the graphics subsystem at these rates requires a bandwidth of 236 MB/s (3.9x60). This figure also represents the sustained, not the peak, requirement and also does not take into consideration the transfer of additional data necessary for a scene such as 3D geometry or even additional textures. To make a system acceptably interactive when using texture mapping, clearly a significant amount of data needs to be moved. From a system architecture viewpoint, two approaches are typically adopted to solve this problem.

The first approach is to use dedicated texture memory on the graphics board where textures are downloaded and stored. This has the advantage that the data can be contained within the graphics subsystem and thus does not invoke the transfer from memory during each frame. Indeed, this is the way adopted by most traditional computer architectures. The disadvantage with this approach is that texture memory local to the graphics subsystem is typically expensive because it is high speed. When texture mapping is not required, therefore, this additional cost is not being utilized. There is also only a finite amount of dedicated texture memory available, so when using a large number of textures it's likely that some will not fit in local texture memory and consequently will have to reside in main memory instead. This clearly, pushes up the system bandwidth requirements. These can in turn create more complexity in application programs since additional functionality may be desired or in fact required to cache textures on the graphics board efficiently .

An alternative approach, and one that is also being promoted by Intel through its AGP bus specification, is to utilize main memory to store textures. Since there is typically a significantly greater amount of main memory than there is dedicated texture memory, this approach potentially offers significant cost savings. Simply by adding more main memory, the capacity to store textures can be increased, and typically main system memory is noticeably less expensive than dedicated texture memory. Also, when texture mapping is not required main memory can be used for other purposes. Unfortunately, though, as the above calculation clearly demonstrates, this approach relies heavily on a significant bandwidth between main memory and the graphics subsystem. The AGP bus is dedicated purely to graphics and thus goes some way to address this particular problem. It should be noted, though, that the AGP specification does not actually preclude it being used for other devices as well as graphics.

Given that AGP 1x, 2x, and 4x have peak theoretical bandwidths of 266 MB/s, 533 MB/s, and 1066 MB/s, respectively, it is easy to see why most high-perfor-

mance graphics cards implemented using AGP typically contain dedicated local texture memory. Indeed, AGP bandwidth figures quoted here are actually peak, and even though texture transfers allow rates close to peak to be achieved, the actual achievable sustained transfer rates will be slightly less than these figures show. When transferring other graphics data such as vertices and normals, the transfers are often more bursty in nature, and in such cases sustained AGP transfer rates are significantly lower than those given above. So although AGP aims to address the problem of designing a system to be low cost while still being able to perform sophisticated features such as texture mapping, it still has noticeable limitations.

The architecture of the Silicon Graphics O2 system addresses the problem by combining the best features of the above two approaches. Combining the function of the memory controller and rendering engine into a Memory Rendering Engine (MRE) (see Figure 3) effectively moves the graphics card significantly closer to both memory and the CPU. Because of this the bandwidth between the MRE and memory can be significantly increased to its current rate of 2.1 GB/s. Clearly, this is also a dedicated connection. The bandwidth between the MRE and CPU is less than this; however, by appropriate design of the graphics functionality in the MRE the need for the CPU to process data during graphics is reduced and thus the bandwidth of this bus does not become a limitation. Clearly for texture mapping, the 2.1 GB/s connection between memory and the MRE is the link over which textures will be transferred during graphics. When compared with the 960 MB/s (@120 MHz) UPA64S bus in the Sun Ultra5 and Ultra10 as well as the 266 MB/s, 533 MB/s, and 1066 MB/s offered by AGP 1x, 2x, and 4x bus, the O2 bandwidth is clearly significantly larger.

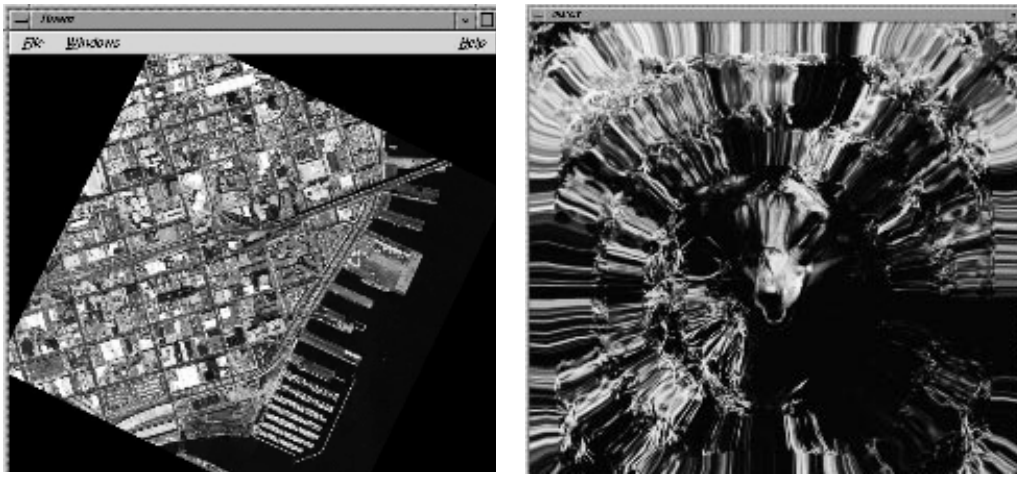
This larger bandwidth doesn't directly equate to overall graphics performance, however, since a large element of overall graphics will relate to geometry processing capability as well as the effects of other activities that occur during dynamic graphics within an application. To this end many of O2's competitive systems, such as the Ultra5 and Ultra10, have significantly more geometry processing capability and in many situations may appear faster. Focusing on pure texture-mapping activity, however, shows O2's strengths. Unfortunately, in published product literature, e.g. [7] and [8], as well as the industry-standard graphics benchmark results [9], there is little data relating to Sun's Elite3D graphics texture-mapping performance.

Image Processing

Image processing is a significant requirement in both the DCC and medical imaging markets. Many applications such as Adobe Photoshop and Adobe Premier are frequently used to perform many sophisticated imaging functions. At a low level, however, all these applications rely heavily on the ability of a system to move data to and from the CPU and memory with low latencies.

O2's architecture is ideally suited to these types of operations due to the close, high-bandwidth link between main memory and the MRE as well as the Image Compression Engine (ICE). The ICE offloads the CPU when performing imaging operations, which significantly improves performance. From a user perspective this doesn't equate to an increase in productivity unless such performance can be utilized transparently. To that end the ICE is designed to accelerate the industry-standard OpenGL imaging extensions and thus is able to offer such functionality to the user in a transparent way.

Unfortunately, there aren't many industry-standard benchmarks related to these activities, and it is difficult to show O2's strengths through benchmark results. By way of demonstration, however, the Roam program included in the standard demo suite on most Silicon Graphics machines provides a very good example of the power and facilities available on O2. The program allows a 1Kx1K image of the San Francisco piers generated from a satellite photograph to be rotated, translated, and zoomed in perfect real time. Through its architecture, O2 is ideally suited to more sophisticated image-processing activities. The Distort demo is also a good example of the power of O2's architecture, remembering that the system lies within the low-cost UNIX market space. The Distort demo presents a photograph that is texture mapped onto a polygonal surface. The surface can be distorted by either introducing ripples by clicking the cursor or by dragging a point upwards in 3D space. After distorting the image in these ways as it returns to its default flat state, the photograph realistically depicts the effect that would be seen. Figures 6a and 6b show a screen shot from both Roam and Distort, respectively.



Figures 6a and 6b, Images from the Roam and Distort demos, respectively.

Video Processing, Compositing, and Virtual Sets

Low-cost video processing is an area that is increasing in popularity within the DCC market space. Indeed, the idea of the virtual 1st down line presented at the start of this paper is a very good example of compositing. This is by no means the only example of compositing; most readers are probably familiar with seeing the picture behind the news announcer changing to represent the current story. This again is an example of compositing. Many readers may be familiar with the concept of virtual sets, where an actor or announcer is recorded against a blue or green screen and then the resulting live action is combined with a computer-generated scene, taking into account the movement of the person within the overall scene. All these examples show that the use of both compositing and virtual sets is gaining widespread use and the results are becoming increasingly more commonplace.

Texture mapping, image processing, video processing, compositing, and virtual sets rely on low-level system features that represent key strengths of O2's architecture. For example, the close coupling of the Display Engine (DE), MRE, and I/O Engine (IOE) allow real-time uncompressed video to be saved to disk. The features built into the IOE provide native support for non-square pixel format and square pixel format conversion in real time, allowing incoming video signals to be displayed with the correct aspect ratios. The IOE also provides support for real-time color space conversion, allowing incoming video signals to be processed and stored in 4:2:2YcrCb format and then converted in real time to 8-bit per component RGB format for use as textures. When this capability is combined with 3D graphics, it presents a very powerful set of features, which are unrivaled by Sun or any of the other competitors at O2's price point. These features are integrated into the standard OpenGL imaging extensions, thereby allowing applications to take maximum advantage with minimal effort. Adobe Premier's transition and 3D special-effects plug-ins are an example of one such application.

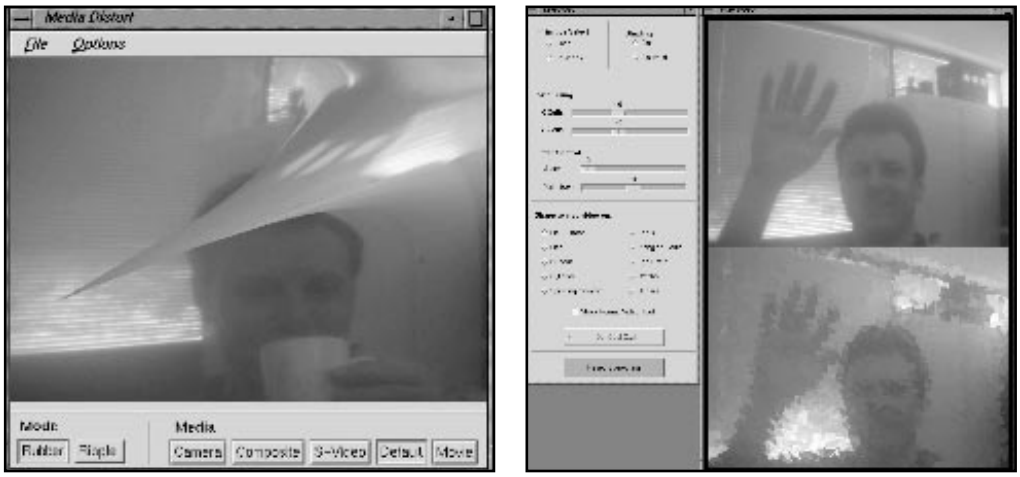
As shown earlier, O2 was deliberately designed so that the video processing components were de-coupled from the IOE and MRE functions. This allows flexibility in configuration to meet customer needs—not everyone will want video in or out capability. What it does mean, though, is that the sophisticated video processing capabilities are available to everyone. To that end if video data originates from disk or network, such as an MPEG stream, then any O2 is capable of using it as a source for texture maps. This is a distinct advantage over competitive systems.

Integrating image and video processing functions into the core components of O2's architecture and closely coupling these components offers several benefits that translate to cost savings when compared with traditional ways of incorporating such features. For example, because the IOE and ICE are closely coupled to main memory, it removes the need for local buffers in the video data paths, with no impact on features of performance. This directly translates to end-user benefits on one level of pure cost savings and on another by facilitating such advanced features as being able to use video data for textures. A more traditional approach to adding such facilities includes incorporating them onto a PCI card. As highlighted by the texture-mapping example above, PCI bandwidth makes providing features such as live video textures very difficult.

Sun is promoting media extensions to its instruction set [10], [11], [12], which it claims when combined with its Ultra Port Architecture (UPA) [13] provides a powerful system with the necessary bandwidths to perform digital media facilities. While it is true that Sun's UPA offers respectable system interconnect bandwidths—1.92 GB/s between CPU and UPA and 960 MB/s between UPA and graphics—the UPA architecture does not offer the same advantages as O2. Indeed, using the CPU for digital media processing may yield several major disadvantages from an architectural perspective.

Using the CPU to perform digital media functions such as video processing is likely to result in it becoming a noticeable bottleneck. It will need to perform other tasks, such as executing the operating system and performing some elements of graphics calculations, depending on the nature of the hardware graphics being used. For video processing, very low latencies and guaranteed processing resource are prerequisites. Because the IOE and ICE in O2 are specifically designed to perform such functions, they can easily accommodate both these prerequisites without also being required to perform unrelated activities, such as running the operating system. Performing the same functions as the IOE and ICE on the CPU through media extensions makes accommodating both these prerequisites considerably more difficult due to the time-slicing nature of how a CPU operates. Additional CPUs may potentially appear to offer a solution; however, apart from increasing cost, it should also be noted that with the UPA architecture, all CPUs will share the 1.92 GB/s to the UPA switch. This may lead to contention, because processing video data on the CPU will typically result in several read and write operations. All three alternatives have to fight for the bandwidth across the connection between CPUs and the UPA switch. Even though the bandwidth between the IOE and MRE is only 533 MB/s, the IOE performs all necessary video conversion as the data arrives, and thus the bandwidth is more than adequate to transmit processed data through the MRE and into memory. Likewise, although it may appear that the bandwidth between the CPU and MRE could be a limitation compared with 1.92 GB/s with Sun's UPA architecture, for applications such as video processing and texturing the MRE operates on data directly in memory. The bandwidth of this connection is 2.1 GB/s, which is clearly significantly higher than that of the UPA. As a result, performing advanced digital media features such as video texturing is quite possible with every O2 system.

This section has tried to provide a sample of the benefits of O2's architecture and features afforded to the video processing end-user. Inevitably, though, because the list of such benefits is very long, there isn't enough space to cover all of them. For a more complete list, as well as further details, the reader is invited to refer to [14]. Good examples of the synergy these feature can yield when applied to video processing are the VideoDistort and the polyVideo demos which are included on the standard Silicon Graphics O2 demo suite. The VideoDistort demo shows how live video can be used as a texture map onto a 3D surface that can be distorted. Likewise, the polyVideo demo shows the types of 3D effects that can easily be created for transitions between video streams by applying video data as textures to morphing 3D surfaces. Figures 7a and 7b show images from the VideoDistort and polyVideo demos, respectively.



Figures 7a and 7b, Images from the VideoDistort and polyVideo demos, respectively.

Computation Performance and Overall System Throughput

The three previous sections demonstrated the strengths of O2's architecture and features by relating some key graphics and digital media technologies to real-world examples. This section discusses the wider issue of where O2 benefits end-user productivity through computation and system throughput. This is an important aspect of system performance in all market segments, especially when application software architectures are evolving to utilize plug-ins. These plug-ins are often developed by software vendors independent of the application vendor producing the software in which the module plugs in. In some cases, such as Adobe Premier's transition and 3D special-effects plug-ins mentioned above, these plug-ins take advantage of the sophisticated digital media features of O2's architecture; however, most typically rely heavily on the CPU. To this end-incorporating specialized functionality into the MRE, IOE, ICE, and DE in O2's architecture not only delivers better overall system performance for specialized features, but also noticeably reduces the load on the CPU. This allows it to dedicate more time to perform additional computations that may be required.

The computational requirements of all users typically won't be the same, and likewise all users typically won't have the same constraints on factors, such as overall system cost, for example. To accommodate these varying needs, O2 incorporates two CPU families: the first family is based on the MIPS R5000 and now the QED RM5200 CPUs, and the second is based on the MIPS R10000 and now the MIPS R12000 CPUs. Both families offer specific characteristics and features, and some of these will be covered in more detail in the next section comparing the new and existing CPUs in both families. To understand the benefits afforded through O2's architecture, though, it is useful to have an appreciation of the benefits both CPU families bring and the relative positioning. The rest of this section presents such an appreciation as well as positioning O2 against some competitive systems.

The industry-standard SPECint95 and SPECfp95 test suites represents a cross-section of applications, exhibiting both integer and floating-point behavior, respectively, and is useful to characterize CPU performance. Figure 8 shows the relative performance of the QED RM5200 300 MHz– and MIPS R12000 300 MHz–based O2 systems.

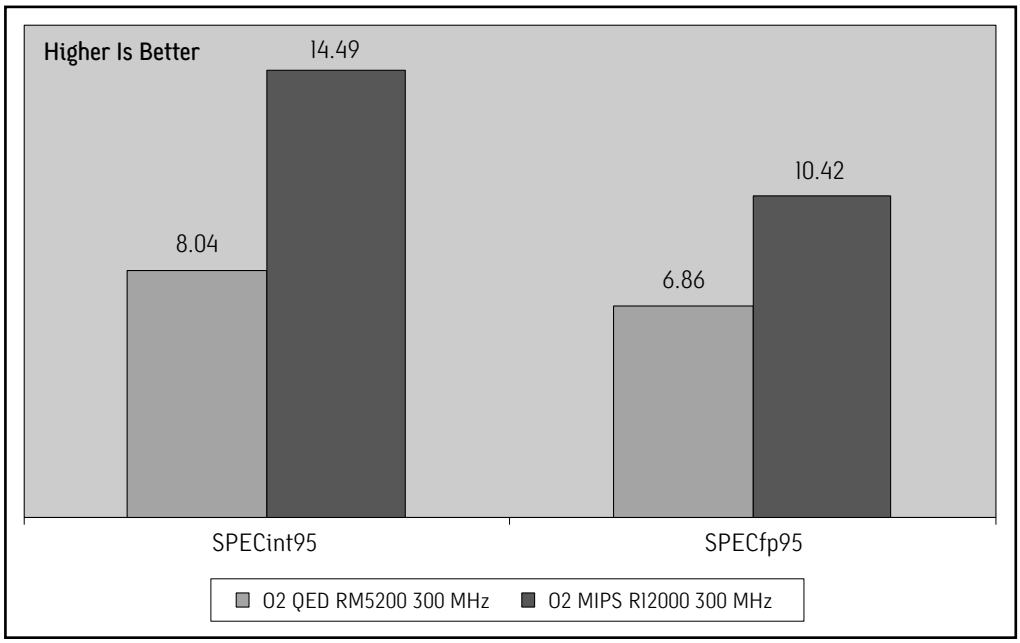


Figure 8, Relative SPECint95 and SPECfp95 performance of QED RM5200 300 MHz– and MIPS RI2000 300 MHz–based O2 systems.

Clearly, the results show that compared with QED RM5200, MIPS RI2000 exhibits stronger integer and floating-point performance. When workloads involve applications demanding significant amounts of computation, MIPS RI2000 would therefore typically be the CPU of choice. This doesn't, however, convey the whole story regarding the relative performance of QED RM5200– and MIPS RI2000–based O2 systems.

Like MIPS R5000, the QED RM5200 microprocessor was designed to be a low-cost 32-bit CPU, taking advantage of the MIPS IV instruction set. The MIPS IV instruction set includes many instructions, such as multiply-add. As will be covered in more detail in the section entitled An Illustration of the Synergy between O2 CPUs and Architecture for Graphics, the MRE utilizes the CPU for appropriate tasks in the OpenGL graphics pipeline. Instructions such as multiply-add are particularly suited to this type of calculation because they save a significant number of CPU cycles compared with performing such operations individually. The graphics libraries supplied as standard on QED RM5200–based O2 systems, and previously MIPS R5000 systems, have been optimized to take advantage of such MIPS IV instructions with noticeable improvements in graphics performance. To illustrate this point, Figure 9 shows the relative graphics performance of QED RM5200– and MIPS RI2000–based O2 systems.

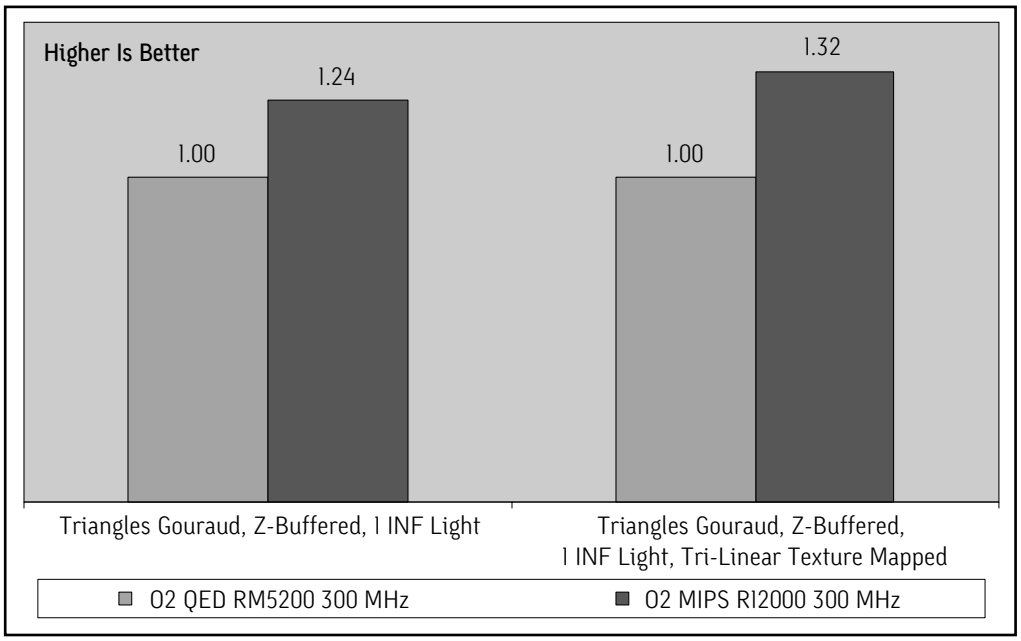


Figure 9, Ratio of graphics primitive performance of QED RM5200 300 MHz– and MIPS R12000 300 MHz–based O2 systems.

Clearly, QED RM5200– based O2 systems deliver good graphics performance relative to MIPS R12000–based O2 systems and provide a strong low-cost entry point to the sophisticated digital media features afforded by O2's architecture. Indeed, many frequently performed tasks in DCC and AEC applications typically yield integer-dominated behavior at a low level, and hence the power of QED RM5200–based O2 systems more than satisfies many user needs. This strength-combined with the very strong client-server scalability of the IRIX operating system, presents a very attractive solution to many customers. When customers require stronger computational performance, such as when applications typically yield more floating-point-oriented behavior, MIPS R12000–based O2 systems are preferred.

Since 1993 the MIPS microprocessors–based Silicon Graphics systems have led most of the industry by being 64-bit addressable. Likewise, in 1996 Silicon Graphics' IRIX operating system was ahead of most of the industry in providing a full 64-bit operating system across its entire product line and is the same operating system that runs on both QED RM5200– and MIPS R12000–based O2 systems.

This scalability of the operating system across the entire product line actually makes O2 systems very attractive to many customers as a development machine. Although this feature doesn't typically take advantage of any of the digital media capabilities of an O2 system, it does offer a very good platform on which to develop applications that would typically run on Silicon Graphics' very high-end graphics and supercomputer systems.

Given the positioning of QED RM5200– and MIPS R12000–based O2 systems, how do they compare with competitive systems? Figures 10a and 10b show the performance of both QED RM5200 300 MHz– and MIPS R12000 300 MHz–based systems relative to competitive systems [15], [16].

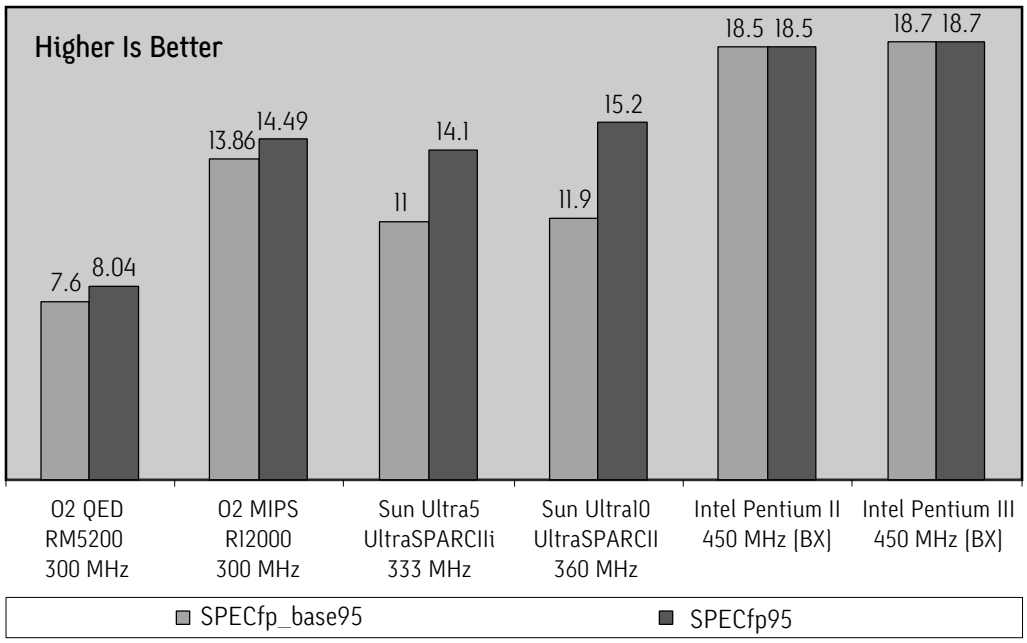


Figure 10a, Relative SPECint95 performance of competitive systems and QED RM5200 300 MHz– and MIPS R12000 300 MHz–based O2 systems.

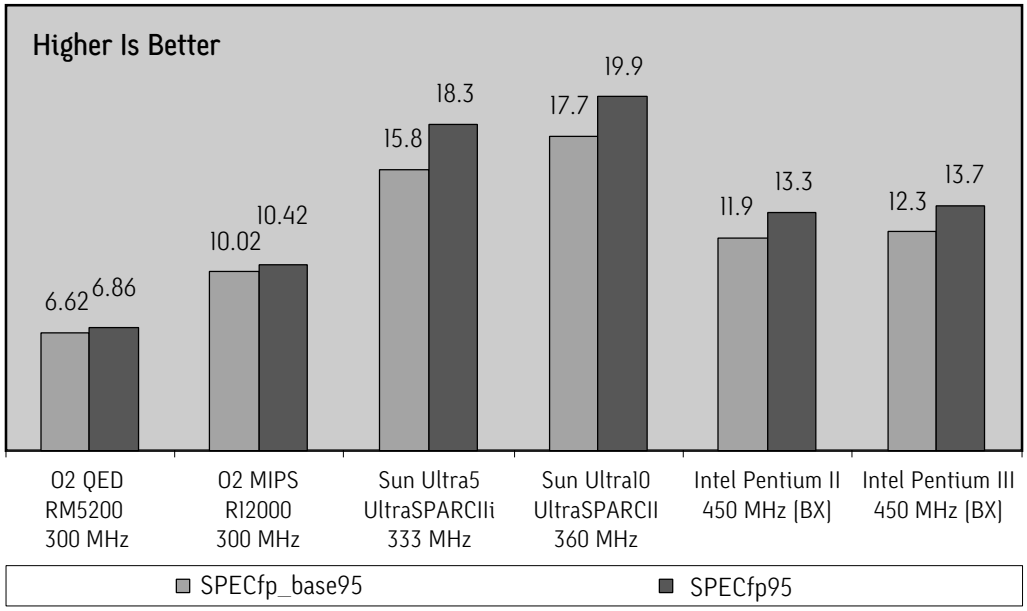


Figure 10b, Relative SPECfp95 performance of competitive systems and QED RM5200 300 MHz– and MIPS R12000 300 MHz–based O2 systems.

Based on the SPEC results shown in Figures 10a and 10b, it would be tempting to think that O2 offers few advantages over the Sun Ultra5 and Ultra10 systems and Intel-based systems. While it is true that the Sun as well as Intel-based systems demonstrate performance advantages in SPECint95 and SPECfp95 results, it is not true that this will be representative of application performance. To illustrate how large the difference between SPEC and application performance can be, Figure 11 shows the performance of a MIPS R10000–based O2 system compared with Sun Ultra5 and Ultra10 systems on the 145.fpppp test, which is part of the SPECfp95 suite. These results clearly show O2 to have a very distinct advantage even compared with the comparable Sun systems.

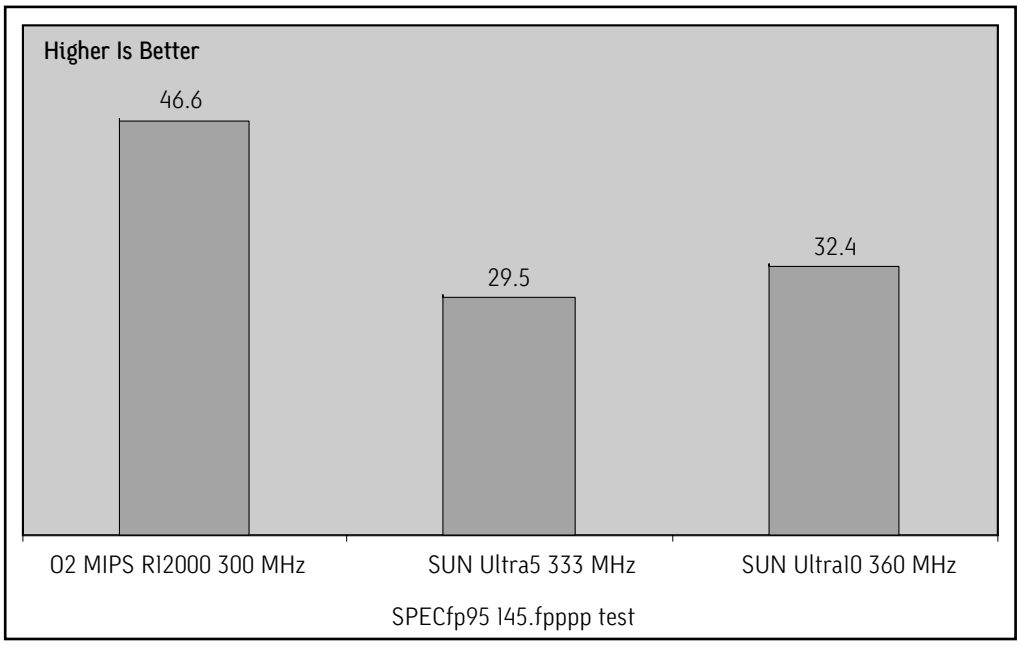


Figure 11, Performance of MIPS R12000 300 MHz– based O2 system compared with Sun Ultra5 and Ultra10 systems for SPECfp95 145.fpppp test.

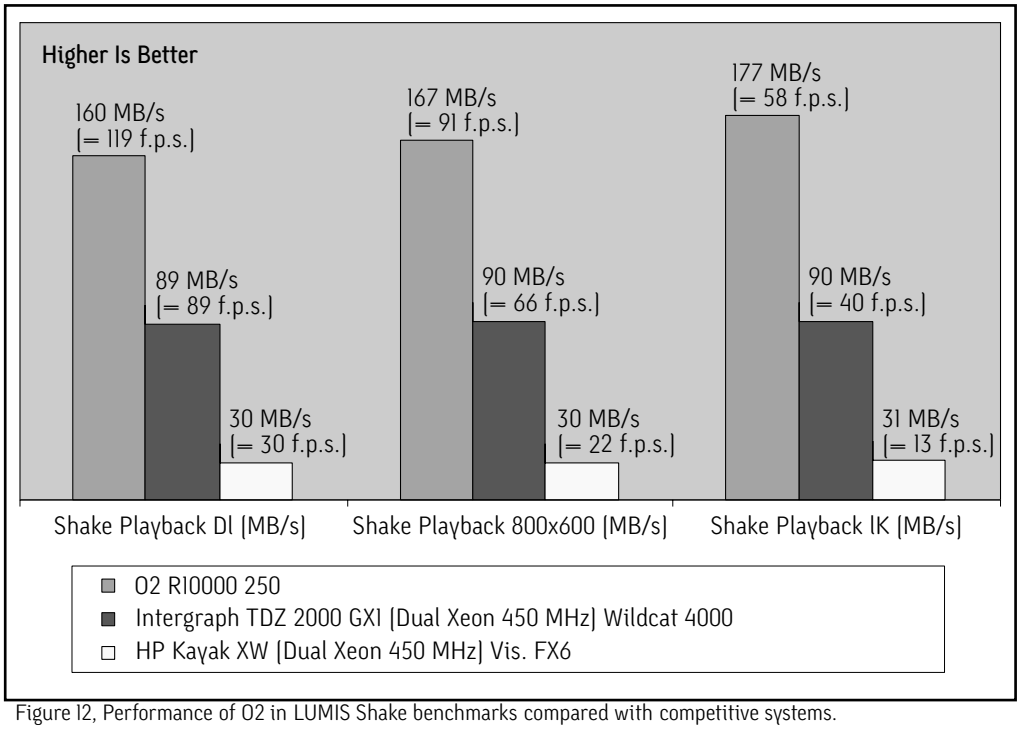


Figure 12, Performance of O2 in LUMIS Shake benchmarks compared with competitive systems.

To further illustrate the same point, Figures 13 and 14 show the performance of a MIPS R10000–based O2 system compared with competitive Intel CPU-based systems measured using the Shake and Maya Render benchmarks run by LUMIS Inc. [17] and [18]. Again, a different result is seen, as would have been expected from the SPEC results alone, which clearly show how well O2 performs on many real applications.

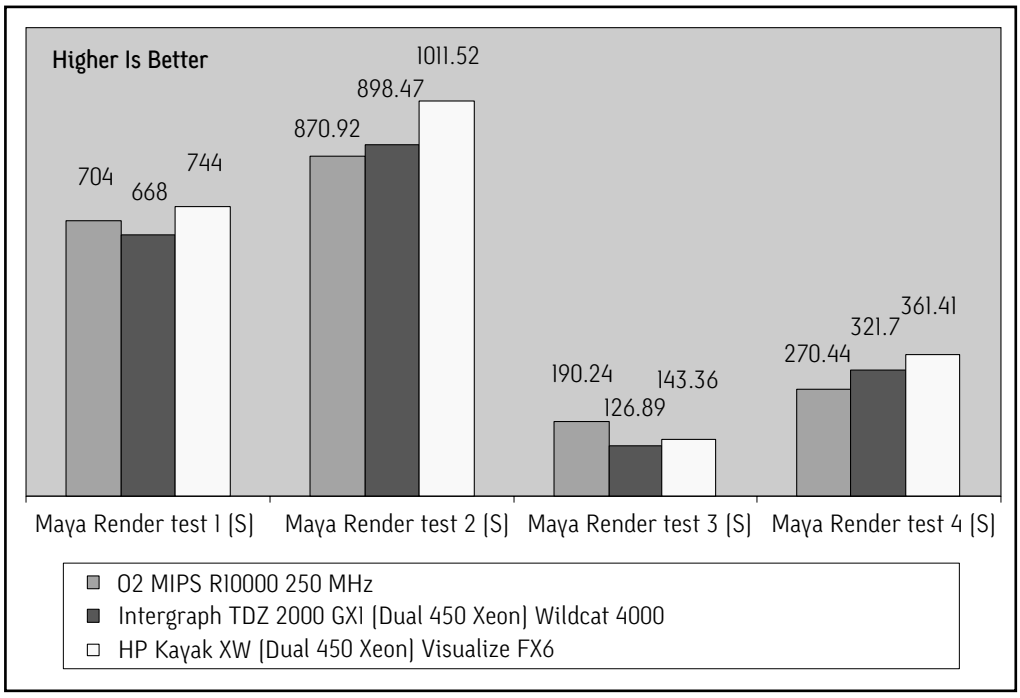


Figure 13, Performance of O2 in LUMIS Maya Render benchmarks compared with competitive systems.

The LUMIS Shake benchmark measures a system's ability to replay movie data from disk, which is an activity frequently performed in digital editing markets and one particularly suited to the strengths of O2's architecture. Clearly, on such activities O2 demonstrates a very significant advantage above high-end competitive systems. Likewise for the more computationally dominated Maya Render benchmark, the MIPS R10000 250 MHz–based O2 system shows comparable performance with the same competitive systems and an 18% advantage over its nearest rival, which uses dual Pentium II 450 MHz Xeon processors.

This is clearly not intuitive from the slight performance disadvantage the same MIPS R10000–based O2 system shows in SPECint95 and SPECfp95 relative to a single Pentium II 450 MHz Xeon processor. Typically, real-world applications highlight the hidden strengths of the O2 system. Combined with the features afforded through its architecture that are unsurpassed by competitive systems, O2 presents a very compelling solution.

Comparison of New and Existing O2 CPUs

This section presents detailed technical information on the two new CPUs available in O2 systems. The first subsection describes the MIPS R5000 and QED RM5200 microprocessors, while the second subsection describes the MIPS R10000 and MIPS R12000 microprocessors. The third subsection covers some of the motivations for the architectural differences between the QED RM5200 and the MIPS R12000 systems.

The section assumes some knowledge of microprocessor architecture and design, and unfortunately there isn't sufficient space in this paper to provide background information. To this end, the reader is referred to [19] and [20] as good sources for general information on these topics. The white paper entitled An Illustration of the Benefits of MIPS R12000 and OCTANE System Architecture [21] also provides some background information on microprocessor architecture and is referenced in the subsection below discussing differences between MIPS R10000 and MIPS R12000.

Description and Comparison of MIPS R5000 and QED RM5200

Figure 14a details the QED RM5200 architecture and pipeline, respectively.

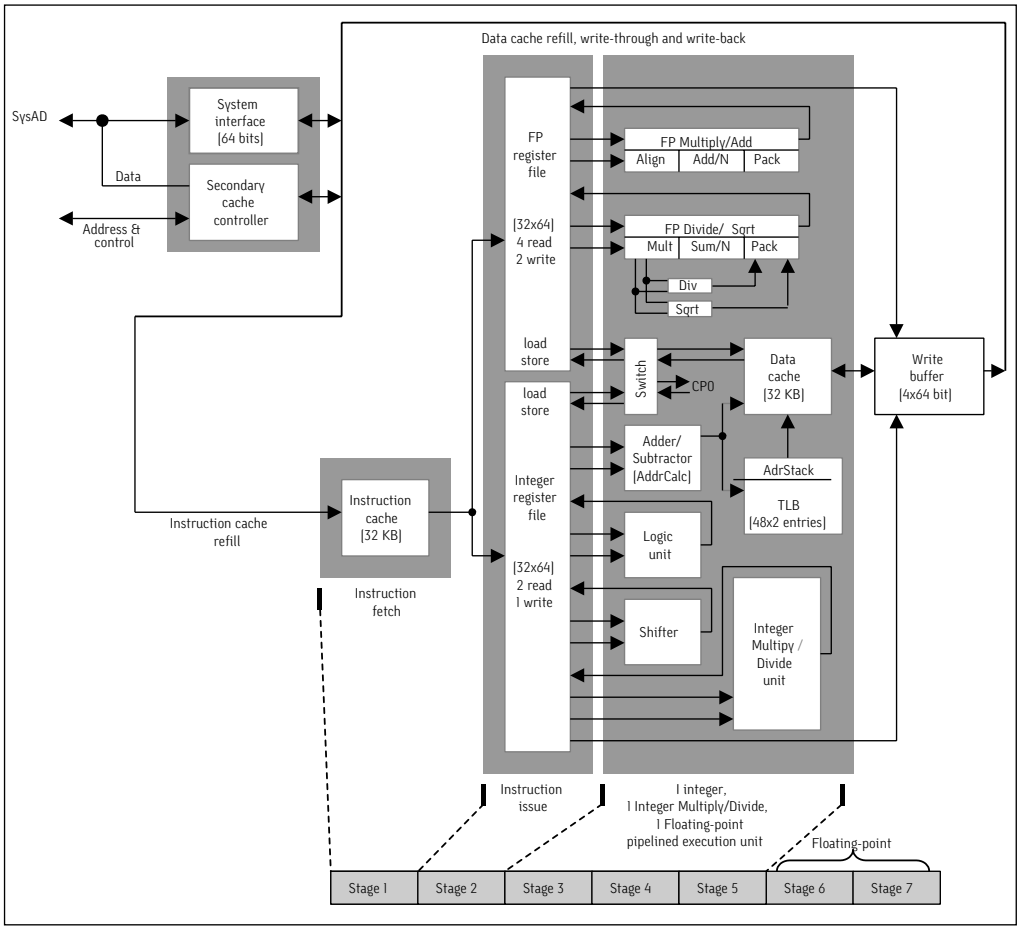


Figure 14a, QED RM5200 architecture.

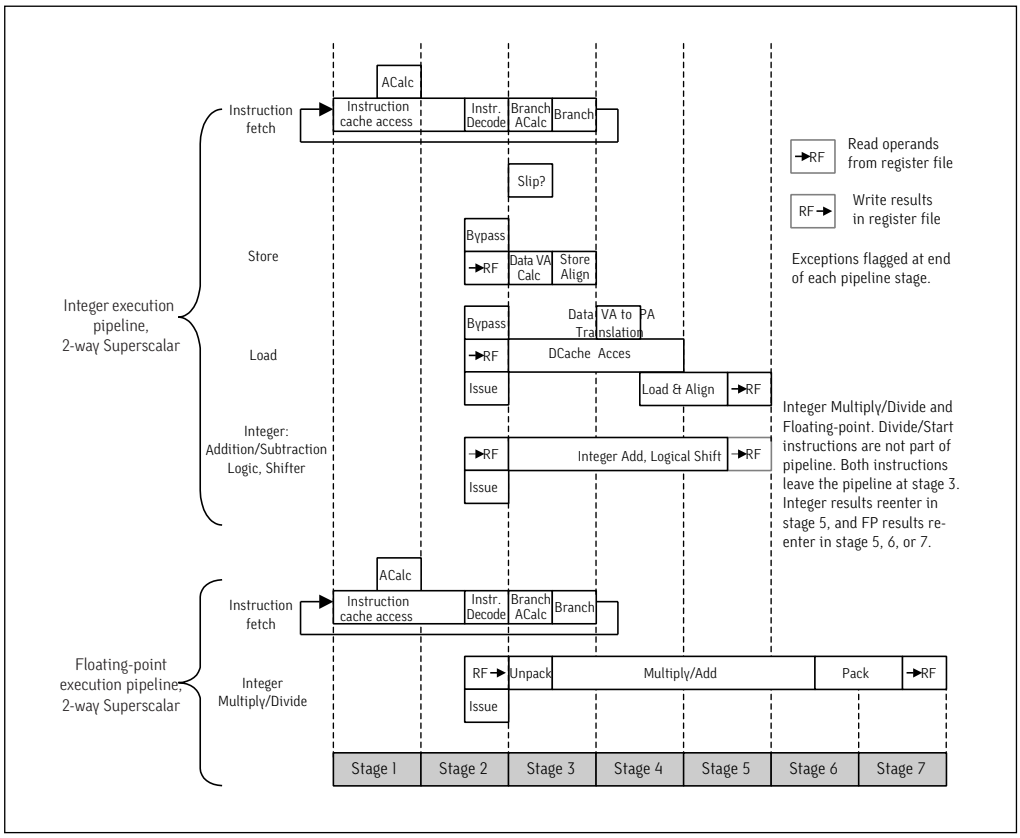


Figure 14b, QED RM5200 pipeline.

Both MIPS R5000 and QED RM5200 have a 32 KB primary instruction and a 32 KB primary data cache. Both primary caches are on-chip and are two-way set-associative. The secondary cache is external, or off-chip, and is also two-way set-associative with sizes of 512 KB, 1 MB, and 2 MB. On O2, both MIPS R5000 and QED RM5200 have a 1 MB secondary cache. Two-way set-associativity offers a good solution to the trade-off between chip complexity and maximizing reuse of cached data, which in turn benefits application performance.

Both MIPS R5000 and QED RM5200 perform secondary cache data writes over the SysAD, or system bus. Secondary cache addresses, as well as some control data, have a separate bus. The motivating factor for this design feature is to reduce chip complexity, and in turn size and cost, by simplifying the secondary cache controller design. Since applications tend noticeably to miss secondary cache, sharing the SysAD bus with secondary cache data may potentially compromise system performance due to increased bus contention. Separating the addresses and control data away from the SysAD bus, however, allows overall application performance to be maintained while retaining the benefits of reduced complexity and costs.

The system interfaces of both MIPS R5000 and QED RM5200 are non-overlapping. This means that only one outstanding request is allowed, and this request must be serviced before another can be issued. Since this could affect secondary data cache transfers, and consequently application performance, both microprocessors also buffer external main memory writes in a buffer that can store four 64-bit data and address pairs. This allows the processor to continue executing after issuing a memory update and typically improves application performance. During uncached write-through stores, the write buffer increases performance by de-coupling SysAD bus transfers from the instruction execution stream. Again, this will yield improvements in application performance.

One of the key differences between the MIPS R5000 and the QED RM5200 is the supported clock rates. As is common on most modern microprocessors, both QED RM5200 and MIPS R5000 use Phase-Locked Loop (PLL) circuitry to multiply an externally supplied clock signal. For MIPS R5000 the maximum speed of the external clock signal and the multipliers to generate the internal clock rate are restricted to 2, 3, and 4, and as a result the maximum internal clock rate of MIPS R5000 is 200 MHz. On QED RM5200 the range of clock signal multipliers is expanded to 2, 2.5, 3, 3.5, 4, 4.5, 5, 6, 7, 8, and 9, which clearly yields a greater range of internal versus external clock speeds. This in turn allows more flexibility when incorporating QED RM5200 into O2 and ultimately facilitates a better match with user application performance requirements.

For both MIPS R5000 and QED RM5200 the SysAD speed NO2 is set to 100 MHz. Given the greater flexibility in internal clock rate on QED RM5200, this may appear as a limitation. It isn't so. The significant primary and secondary cache sizes, coupled with the performance advantage afforded through the write buffer, have shown the 100 MHz maximum SysAD bus speed not to be a limiting factor.

The Translation Look-aside Buffer (TLB) on both MIPS R5000 and QED RM5200 applies to both data and instructions and is referred to as the Joint TLB (JTLB). On both microprocessors it is direct mapped and accommodates varying page sizes between 4 KB and 16 MB on a per-page basis. Being able to adjust page size allows increased flexibility in meeting the system requirements imposed through running many varied applications.

The JTLB can store 96-page translations organized as 48 dual entries. Organizing the JTLB in this fashion significantly reduces the area on the microprocessor chip dedicated to the TLB, and in turn saves both in chip area and overall power consumption. Since nearly every instruction executed by the microprocessor typically incurs a page translation, maximizing the size of the table to 96 overall entries will improve application performance by allowing more data and instruction page translations to be cached in the JTLB, thereby avoiding expensive memory accesses. On RM5200 there is also a 2-entry Instruction TLB (ITLB) and a 4-entry Data TLB (DTLB). These further improve application performance, because when instruction and data page translations hit both the ITLB and DTLB simultaneously, both data and instruction address translations can occur in parallel. If such translations should miss either the ITLB or DTLB, they are both transparently filled from the JTLB.

Both MIPS R5000 and QED RM5200 implement a 5-stage integer pipeline and a 7-stage floating-point pipeline. Both the integer and floating-point execution units incorporate individual instruction fetch logic and consequently are both dual issue, allowing one floating-point instruction and one integer instruction to be issued per cycle. Clearly, for codes that contain an even mix of floating-point and integer instructions, this greatly assists application performance.

The integer unit includes an adder/subtractor execution unit, which also calculates addresses; a logic execution unit; a shift execution unit; and a separate integer multiply/divide unit. Even though the integer multiply/divide unit has a longer latency compared with the other units, once an instruction is issued to it, subsequent instructions can be issued to the other units and thus the longer latency has little direct impact on overall execution. In a similar way, the floating-point unit is divided into a multiply/add unit and a divide/square root unit. Although the latency of the divide/square root unit is again longer than that of the integer queue, once an instruction is scheduled to it the processor can then schedule subsequent instructions to the multiply/add unit. As a consequence, the longer latency has less effect on overall throughput and application performance. There are 32 general-purpose integer and 32 floating-point registers on both MIPS R5000 and QED RM5200. Registers are where the operands associated with instructions are stored while they are executing on the microprocessor. Having more will clearly allow more instructions to be executing in the microprocessor at one time; however, this needs to be balanced with other features of the processor, such as the number of instructions that can simultaneously execute, as well as the number of outstanding memory accesses. For both MIPS R5000 and QED RM5200, 32 integer and 32 floating-point registers were found to be suitable for the overall design of the microprocessors. Both MIPS R5000 and QED RM5200 also have two special-purpose 64-bit registers for the integer multiply/divide execution unit. These special-purpose registers facilitate two instructions beyond the MIPS IV ISA instruction set: integer multiply-accumulate and three operand integer multiply; however, both these instructions are primarily targeted at the embedded processor market and are not utilized by typical O2 applications.

As mentioned above, since secondary cache data transfers share the SysAD bus with memory accesses, this could potentially yield an increase in bus traffic and contention, impacting application performance. By default, therefore, both MIPS R5000 and QED RM5200 use a write-back cache protocol to reduce system bus traffic and eliminate potential delays waiting for memory accesses. In some situations, however, it is desirable to change the cache protocol since caching is undesirable and may in fact cause an overhead to overall throughput. One such example relevant to O2 is when addressing screen buffer data. To support this, both MIPS R5000 and QED RM5200 offer write-through protocols that can be selected by software on a per-page basis.

The four types of protocol supported by MIPS R5000 and QED RM5200 are defined as follows:

1. Uncached: Sections of memory using an uncached protocol will not access cache during reads and will not update the cache during writes.

2. Write-back: When reading data, including load operations and instruction fetches, a write-back protocol will search each cache level in turn. If no hits occur the data will be fetched from main memory. For writes, i.e., store operations, each cache level is checked to see if the location is included in a cache line. If a hit occurs, the location is updated and data is written to main memory and the cache line marked dirty. If no hits occur in cache, then the cache line is loaded from memory and then written with no update to cache.

3. Write-through with write allocate: Using this protocol, reads will search the primary cache and go to main memory if the data is found not to be resident. Using a write-through protocol, data is not stored in secondary cache. On data writes, again the primary cache is searched and if the location is found to be resident, the value is updated both in primary cache and main memory and the write-back bit of the cache line is also left unchanged. No writes occur to the secondary cache. If the data is not resident, then the cache line is loaded from memory into primary cache, updated, and written back with no update to secondary cache.

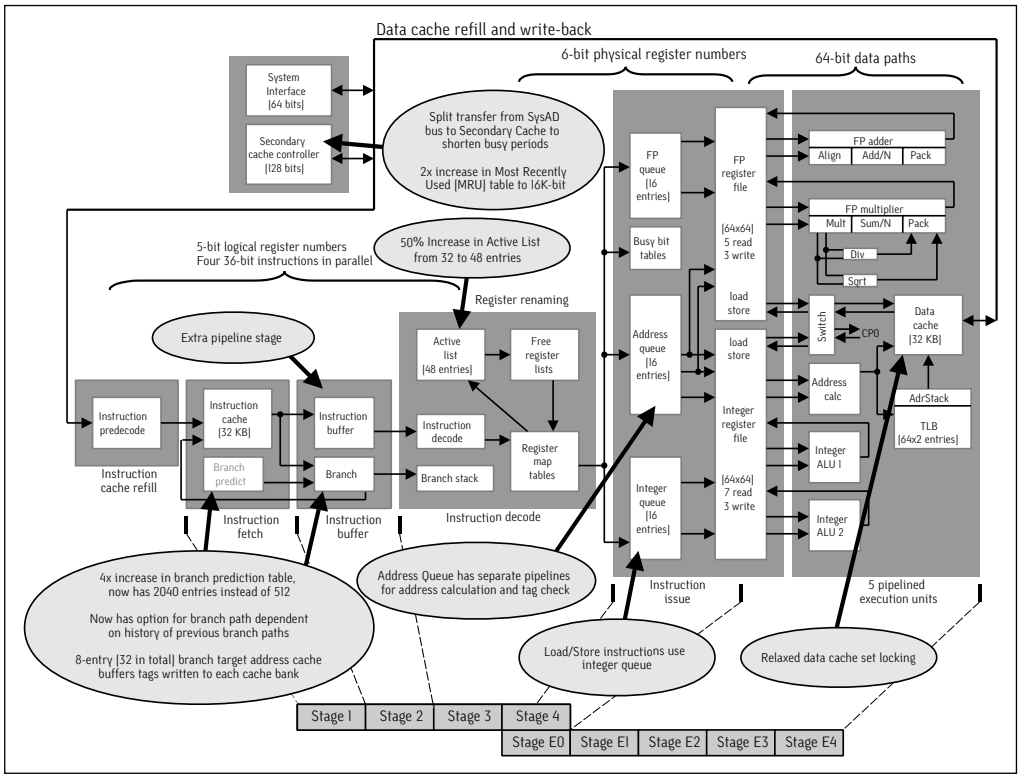
4. Write-through without write allocate: This protocol is the same as the write-through with write allocate protocol except that when writing data, if the data is not resident in primary cache, then the value is read in and written back to memory without loading the data into primary cache.

The four cache protocol alternatives allow significantly increased flexibility when designing a system such as O2 so that it can meet the demands of many varied applications. The use of main memory for the frame buffer and texture memory places significantly different requirements on a system to obtain maximum performance compared with, say, an application that generates rendered images. Being able to choose the appropriate cache protocol avoids unnecessary transfers over the SysAD bus traffic, thereby reducing contention, processor delays, and so on, and maximizing application performance. Furthermore, since the write-through cache protocol can be individually set for each page translation stored in the JTLB and DTLB, and each entry in the JTLB and DTLB can accommodate different page sizes, one JTLB entry can potentially accommodate an entire frame buffer. Clearly, compared with fixed page size JTLB and DTLB entries, this frees up many entries that can be utilized by other executing codes. This in turn helps maximize overall application performance and throughput, which is of obvious benefit to the end-user.

The information in this subsection is intended to be a summary of the MIPS R5000 and QED RM5200 architectures and the key differences. Inevitably some aspects of both architectures don't receive as much information as perhaps they should, so to that end [22] and [23] provide more details on both microprocessors.

Description and Comparison of MIPS R10000 and MIPS R12000

Figures 15a and 15b show the MIPS R12000 architecture and pipelines respectively, highlighting key differences relative to MIPS R10000.



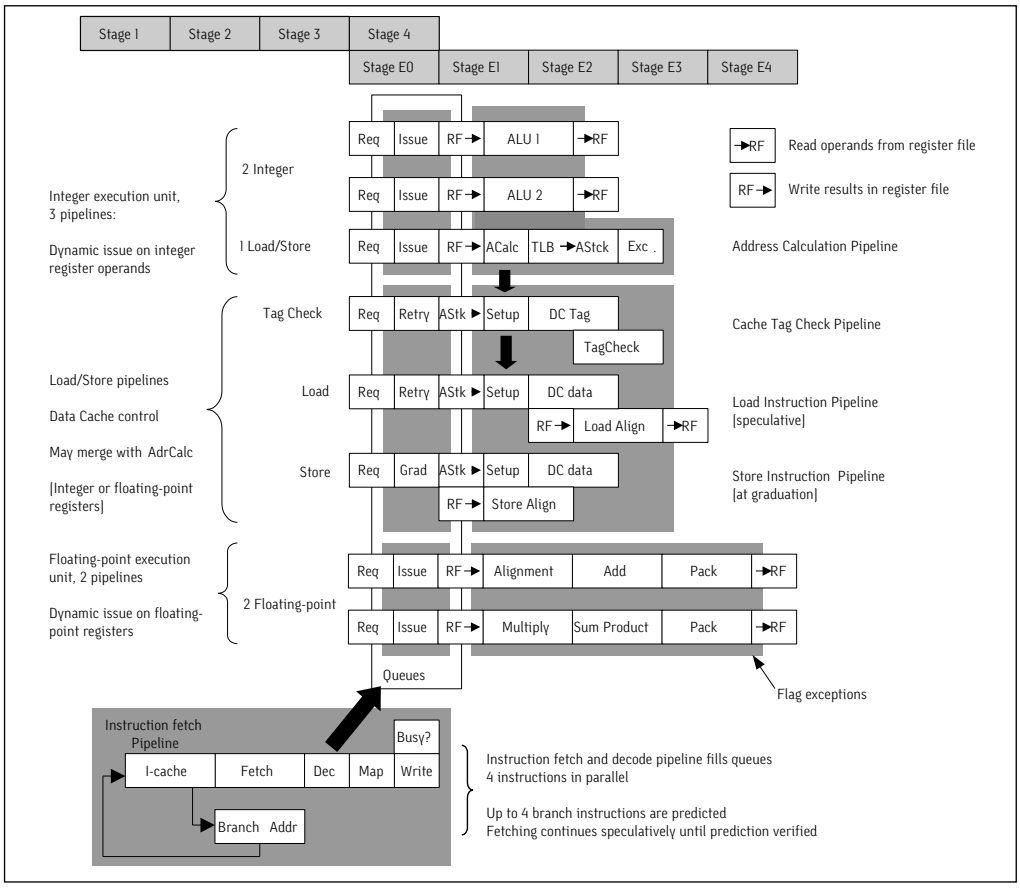


Figure 15a and 15b, MIPS R12000 architecture and pipeline showing key differences relative to MIPS R10000.

As can be seen from Figure 15b, there are several changes when comparing MIPS R10000 and MIPS R12000, and a full description of all of these is provided in [21]. Due to the nature of how the CPU is utilized for graphics in O2's architecture, some of these changes will exhibit more influence and those will be briefly discussed here. Like the change from MIPS R5000 to QED RM5200, the secondary cache size is also constant between MIPS R10000 and MIPS R12000 on O2; any performance improvements are a direct result of processor architecture as well as clock speed differences.

As in OCTANE, the increased active list in MIPS R12000 in O2 will allow more instructions to be executing or pending execution. The resulting increase in speculatively executed instructions will improve application performance. A further benefit on O2 is that since secondary cache transfers share the SysAD bus, an increased active list accommodates more pending load/store operations and is likely to help prevent processor stalls when bus traffic is high.

As mentioned in [21], most large application codes exhibit a significant number of branch instructions, sometimes even up to 15–20% of all the instructions. Stalling the processor pipeline as a result of a mis-predicted branch incurs a very significant penalty, and a four-fold increase in the branch prediction table is likely to yield a very significant improvement application performance. The incorporation of a branch target address cache, which significantly reduces the time taken to obtain a branch address, will add to these benefits. Graphics libraries typically include a significant number of branch instructions to accommodate varying states within applications, and as a consequence the increased branch prediction table length will significantly help. During dynamic graphics instructions are also repeated many times and the locality of reference is typically higher. As a result, the branch target address cache will also noticeably improve performance.

Although the secondary cache address and control signals are separated from the SysAD bus in O2's architecture, the results of such searches are still transferred across the SysAD bus. As a consequence, correct way prediction will slightly decrease this traffic and potentially improve performance through reduced bus contention. As shown in [21], the increase in MRU table in MIPS R12000 noticeably reduces the number of mis-predicted branches. Integrated into the architecture, this change will typically reduce the contention on the SysAD bus and offer performance benefits beyond the direct reduction in time spent checking secondary cache.

Like the previous subsection, this subsection is intended to be a very brief summary of the differences between MIPS R10000 and MIPS R12000. As mentioned above, [21] discusses these details in significantly more detail. Further details and information on both microprocessors are given in [24] and [25].

Summary of Differences between QED RM5200 and MIPS R12000

The previous two subsections presented a summary of the differences between the new and the previous processors in both families offered in O2 systems. This section highlights some of the more significant differences between the two microprocessor family architectures in more detail, explaining the motivations and reasons behind them and how they translate into the positioning of O2 systems.

MIPS R12000 instruction fetch and decode unit allow four instructions to be decoded and scheduled per cycle, which allows the five execution units to be kept as busy as possible. Such features, while significantly adding to the throughput of the processor for many applications, do increase the size, complexity, and cost. Some applications, such as video processing, however, typically yield many similar instructions, such as memory read and writes, and by their nature don't take full advantage of the sophisticated features of MIPS R12000. For such applications, even though QED RM5200 replicates the instruction fetch and decode in both the integer and floating-point queues, and only allows a maximum of two instructions to be executed per cycle, the overall system performance is very high. To a user this is made even more attractive by the significantly lower overall system cost. As a consequence, previously MIPS R5000– and now QED RM5200–based O2 systems offer very powerful system for applications oriented toward video at low cost. Such applications clearly allow full advantage to be taken of the sophisticated features afforded by O2's unique architecture.

Likewise, for computationally intensive applications MIPS R12000 incorporates out-of-order execution to minimize the effects of waiting for memory accesses as well as for other reasons. QED RM5200 does not include this technology, but again this also significantly simplifies its design and lowers the overall cost.

The same applies to other advanced MIPS R12000 features, such as branch prediction. In terms of positioning MIPS R12000 and QED RM5200 O2 systems, the features of MIPS R12000, such as out-of-order execution and branch prediction, can yield significant improvements to overall system throughput for applications that are more computationally intensive. From a user perspective, though, these improvements need to be balanced with overall system cost.

Similarly, the MIPS R12000 employs an MRU table to predict which way of the secondary cache is searched first, which can significantly reduce the time taken to check if data is resident in secondary cache. Since most applications typically exhibit a large number of secondary cache accesses, this can be very beneficial. QED RM5200 doesn't implement an MRU table and relative to MIPS R12000, this may yield a slight overhead during secondary cache searches. With real-world applications, however, many other factors can affect overall system performance. Again from a user perspective, the specific application(s) being used will determine the tradeoff between potentially higher computational performance of MIPS R12000–based O2 systems versus the cost advantages of the QED RM5200–based O2 systems.

Similar reasoning applies to comparisons between features such as the number of registers on MIPS R12000 compared with QED RM5200. Since MIPS R12000 has 64 integer and 64 floating-point registers compared with 32 integer and 32 floating-point registers on QED RM5200, it may appear that MIPS R12000 will have a significant advantage—a greater number of registers allows a greater number of instructions to be executing in the processor. While this is true in principle, the effect it has on application performance will be very dependent on many other factors, including the distribution of instructions between load/store; integer or floating-point instruction, which is very application specific; and other factors arising from specific features of the microprocessor design, such as how many instructions can be scheduled in one cycle. As a consequence, the price performance advantages of QED RM5200–based systems compared with higher potential computational capabilities of MIPS R12000 have to be balanced. In areas such as video processing, the QED RM5200–based O2 systems offer very attractive solutions.

An Illustration of the Synergy between O2 CPU and Architecture for Graphics

Most applications that are well suited to O2's architecture yield a substantial element of graphics activity, and even though the specifics of each application may vary, typically graphics activities translate to drawing lower-level graphics primitives such as triangles or lines. This section explains how MIPS RI2000 and QED RM5200 combined with O2's architecture demonstrate significant benefits in graphics primitive performance. How this translates to application performance is covered by the next section, Translation of Improvements Afforded through New CPUs into End-User Benefits.

Figure 16 shows a schematic diagram of the OpenGL pipeline and how it maps to the key components of O2's architecture.

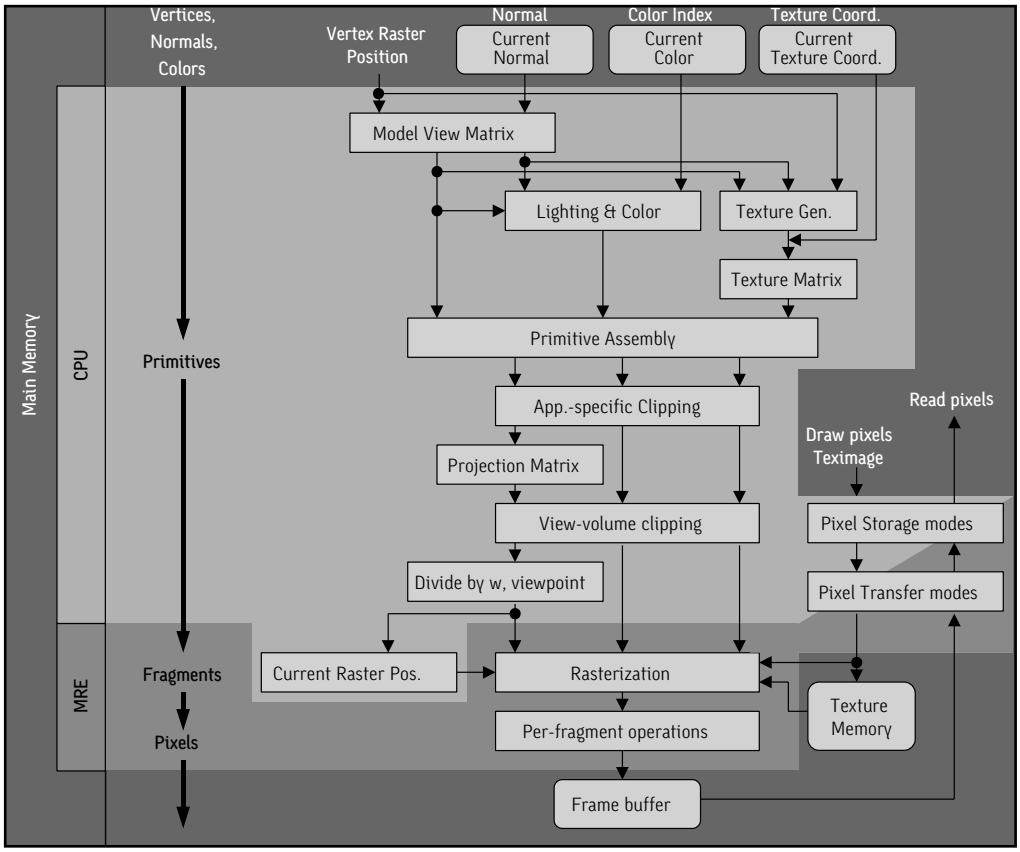


Figure 16, Diagram of OpenGL pipeline and how it maps to the key components of O2 architecture.

As shown in Figure 16, O2's architecture uses the CPU for several stages in the OpenGL pipeline. Vertex data, comprising vertices, normals, and colors, is either individually stored in main memory or within vertex arrays and display lists. Viewing and projection matrix transformations, as well as view volume clipping and triangle setup are performed on the CPU. Subsequent primitive data, such as lines and triangles is sent to the MRE, which completes the rasterization process and performs depth buffering and other pixel-based tests. Finally, the pixel data is stored in the frame buffer, which, as mentioned earlier, is a dedicated section of main memory. In the case of textures, the packing format, e.g., RGB or RGBA, dictates whether they will be unpacked by the CPU and subsequently pushed to the MRE or whether they will be pulled directly from main memory by the MRE.

Since the CPU is such an integral part of O2 graphics, one would expect in most cases an increase in CPU performance to lead to a corresponding improvement in graphics performance. Table 1 shows the performance improvements afforded through QED RM5200 and MIPS RI2000, relative to MIPS R5000 and MIPS RI0000, respectively, for triangle fill-rate and texture downloads. The relative improvements for clock rate, SPECint95, and SPECfp95 increases are also given for comparison.

	Performance of QED RM5200 Compared with MIPS R5000	Performance of MIPS RI2000 Compared with MIPS RI0000
Triangle fill-rate (Triangles/s) (Gouraud shaded, Z-buffered, 1 infinite light)		
1 Pixel	1.57x	1.26x
25 Pixel	1.58x	1.22x
50 Pixel	1.38x	1.01x
Triangle fill-rate (Triangles/s) (Gouraud shaded, Textured, 1 infinite light)		
1 Pixel	1.47x	1.30x
25 Pixel	1.47x	1.31x
50 Pixel	1.47x	1.15x
Texture Image Downloads (Images loaded/s) (64x64, Unsigned byte, mipmapped, no primitive)		
RGB	1.41x	1.34x
RGBA	1.42x	1.24x
Clock Speed	1.50x	1.20x
SPECint95	1.49x	1.20x
SPECfp95	1.16x	1.07x

Table 1, Triangle fill-rate and texture image download performance improvements afforded by QED RM5200 and MIPS RI2000 relative to MIPS R5000 and MIPS RI0000.

Most readers are probably familiar with the fact that objects drawn in three dimensions on a computer screen are typically represented by triangles. The triangle fill-rate results shown in Table 1 measure the speed with which triangles can be sent to the screen after performing additional operations, such as Z-buffering and lighting. The two triangle fill-rate cases shown present results for gouraud shaded, Z-buffered, one infinite light and gouraud shaded, one infinite light, 64x64 trilinear modulated texture cases.

The size with which a triangle appears on the screen clearly affects the number of pixels sent to the frame buffer and thus affects the speed with which a triangle is drawn. When fewer pixels are drawn, the time taken to calculate and send pixels to the frame buffer is reduced relative to performing the geometric transformations on the vertices and normals. In cases where triangles screen area is very small, say, less than 1 pixel, the overall time taken to draw a triangle is typically dominated by the geometry calculations. Such cases are referred to as geometry limited. In cases where individual triangles result in many screen pixels, and may also be textured, then the time taken to draw an individual triangle is typically dominated by the time taken to create individual pixels and send them to the frame-buffer. Such cases are referred to as fill limited. In real life the number of pixels within a triangle is very dependent on both model viewpoint and orientation. In the results above, 1- pixel, 25-pixel, and 50-pixel triangles are included to give an indication of performance over a range of different viewpoints and orientations.

The texture image download tests measure the speed with which texture images can be transferred from memory to the graphics. On O2, because main memory is used for both texture memory and the frame buffer, the path taken by data is slightly different compared with other more traditional graphics architectures, such as Sun Creator3D. On these architectures both texture and frame buffer memory are located on the graphics subsystem. Many readers are also probably aware that there are many different formats by which image data used for textures can be stored, which affects the accuracy and number of bytes used to store the red, green, and blue components and also whether the image includes transparency or not. The RGBA format quoted above is typically used by many applications that utilize texturing. The RGB format has also been included because it uses the CPU to unpack the image data before the texture is sent to the MRE engine and hence the CPU has more potential to influence performance.

The rest of this section describes how the CPU influences O2 system performance for each of the above tests; however, it is worth briefly mentioning the method used to gather the necessary data.

Both MIPS RI0000 and MIPS RI2000 include performance counters that can be used to measure internal performance. These counters measure metrics such as cache misses, mis-predicted branches, and instructions executed, and provide insight into the parts of the processor stressed by executing code. Unfortunately, due to their simpler design, neither MIPS R5000 nor

QED RM5200 include such performance counters, so the examples provided below do not in-clude examples for these processors. It is likely that some of the results described for MIPS R12000 will carry across to QED RM5200, although due to the microprocessor architectural differences many will not.

In [21] the benefits afforded through the architectural improvements of MIPS R12000 relative to MIPS R10000 are demonstrated through an example using the CATIA SolidE function. For this example the CPU time during the test was almost exclusively devoted to the application code, CATIA. The graphics tests used to generate the results shown in Table 1 use application specifically designed to stress aspects of a system's graphics performance. Consequently, this typically results in a significantly greater amount of CPU time spent executing system library code. This is particularly true for O2 since the CPU is utilized for many of the graphics functions. Unlike application code, system libraries typically contain many sections of code that are dependent on a specific processor type, including both MIPS R10000 and MIPS R12000. Whereas [21] demonstrated the influence of MIPS R12000 architecture improvements by showing the differences in recorded processor counter values, a similar approach for the graphics tests given above would be less appropriate due to the significantly greater influence of processor-dependent code. To that end, higher-level statistics such as graduated instructions per cycle are more appropriate and have been used below; however, even then it should be remembered that any differences in code do not allow exact comparisons between architectures.

Triangle Fill-Rate

Figure 17a shows an extract of OpenGL pseudo code taken from the triangle fill-rate test. It purposely doesn't contain all the code executed during the test; however, it is intended to represent the key elements of the test.

```
.....
glEnable( GL_DEPTH_TEST );
glEnable( GL_LIGHT0 );
glShadeModel( GL_SMOOTH );
.....
START LOOP
glBegin( GL_TRIANGLE_STRIP );
glNormal3fv(-0.695387, -0.695387, 0.181309 );
glVertex3fv(-3.43724, -3.99015, -5.89054 );
glNormal3fv(-0.695359, -0.705928, 0.134688 );
glVertex3fv(-3.45947, -4.0145, -5.87967 );
glNormal3fv(-0.684835, -0.69964, 0.203727 );
glVertex3fv(-3.41042, -4.00736, -5.85063 );
glNormal3fv(-0.674367, -0.703943, 0.222919 );
.....
glNormal3fv(-0.450128, -0.16489, 0.877608 );
glVertex3fv(-1.4386, -2.75375, -4.98767 );
glNormal3fv(-0.449494, -0.176747, 0.875623 );
glVertex3fv(-1.45689, -2.77448, -4.9768 );
glNormal3fv( -0.458765, -0.178623, 0.870419 );
glVertex3fv(-1.47405, -2.76471, -4.98318 );
glEnd();
END LOOP
.....
```

Figure 17a, OpenGL pseudo code during triangle fill-rate test.

The first three lines show the settings enabling gouraud shading, Z-buffering, and lighting. During the test, triangles are sent to the graphics as triangle strips, comprising 120 triangles per strip and 10 strips overall. The vertices and normals sent are generated from random numbers and stored before the test measurement commences. During the test the triangle strips are repeatedly drawn for a period of 10 seconds and then, based on the time taken and the total number of triangles, the numbers of triangles drawn per second is calculated.

As the triangle fill-rate results in Table 1 show, the percentage speed for both RM5200 and MIPS R12000 is significantly greater than clock rate for both 1- and 25-pixel cases. This may seem as though we are getting something for nothing; however, being able to perform geometric calculations faster and better cache reuse, especially in view of the use of the SysAD bus for cache data transfer, is likely to yield this result. As the triangle size increases to 50 pixels, fill limitations become more noticeable and, while the QED RM5200 still shows reasonable improvement, the MIPS R12000 yields little speed.

Table 2 shows graduated instructions/cycle, graduated floating-point instructions/cycle, and graduated load store instructions per cycle.

	MIPS R10000 250 MHz	MIPS R12000 300 MHz
Graduated instructions per cycle		
1 Pixel	1.220049	1.293745
25 Pixel	1.243763	1.302638
50 Pixel	1.206619	1.311304
Graduated floating-point instructions per cycle		
1 Pixel	0.344053	0.379708
25 Pixel	0.353285	0.386848
50 Pixel	0.341924	0.394801
Graduated load/store instructions per cycle		
1 Pixel	0.458965	0.493239
25 Pixel	0.467848	0.504176
50 Pixel	0.456738	0.498432

Table 2, Graduated instructions per cycle, graduated floating-point instructions per cycle, and graduated load store instructions per cycle.

Graduated instructions per cycle, including floating-point and load/store instructions, represent the throughput of the processor and provide a measure of how well the processor is able to execute code. Many of the architectural improvements in the MIPS R12000 are specifically intended to improve the processor's ability to execute instructions: The increased active list allows more instructions to be executing or pending execution and increases speculation. The increased branch prediction table typically yields better branch prediction and leads to fewer flushes of the execution pipeline. The increased MRU table typically yields fewer delays when data is resident in secondary cache and allows the processor to be fed with more instructions. As can be seen from Table 2, the graduated instructions, graduated floating-point instructions, and graduated load/store instructions observed during the triangle fill-rate test all increase for the 1-, 25- and 50- pixel triangle cases, showing that the architectural modifications do indeed yield the desired results.

	MIPS R10000 250 MHz	MIPS R12000 300 MHz
L1-L2 bandwidth		
1 Pixel	12.94796	16.53812
25 Pixel	14.36574	18.06257
50 Pixel	17.85962	22.61384
Memory bandwidth used		
1 Pixel	1.622079	2.04869
25 Pixel	1.689756	2.108009
50 Pixel	1.805317	2.227033
Average MFLOPS		
1 Pixel	88.4582	106.545
25 Pixel	87.57669	111.4861
50 Pixel	85.48092	118.4403

Table 3, The L1-L2 bandwidth used, memory-bandwidth used, and average MFLOPS seen during the test.

The L1-L2 bandwidth represents the amount of data transferred between the primary and secondary caches during the execution of a program. It is defined by combining the number of bytes transferred from secondary cache to primary cache with the number of bytes written back from primary cache to secondary cache and dividing the total by the overall execution time. Since MIPS R12000 uses a write-back cache protocol, it represents how well primary cache misses are being satisfied from secondary cache. Since secondary cache misses will cause memory access, an increase in the L1-L2 bandwidth is beneficial to code execution. The increased MRU table on MIPS R12000 typically reduces the time taken when a data location is in secondary cache and thus is likely to result in an improvement in L1-L2 bandwidth. Likewise, increasing the graduated load/store instructions per cycle is also likely to improve the L1-L2 bandwidth.

Similarly, the memory bandwidth used represents the amount of data transferred between secondary cache and main memory. It is defined by combining the number of bytes transferred from main memory to secondary cache with the number of bytes written back from secondary cache and dividing the total by the overall execution time. An increase in the bandwidth typically demonstrates that the microprocessor is being more efficient since it shows that more data is flowing into and out of the processor during execution. Again, increases in all graduated instructions, but especially load/store instructions, would be expected to yield an increase in memory bandwidth used.

MFLOPS (Millions of floating-point instructions) represents the graduated floating-point instructions divided by the overall execution time and is often used as a measure of computing power. An increase in MFLOPS demonstrates that more floating-point instructions are being performed in the same time. An increase in graduated floating-point instructions per cycle would typically be linked with an increase in MFLOPS.

As can be seen from the results shown in Table 3, the architectural features of MIPS R12000 yield increases in both the observed L1-L2 and memory bandwidth used along with the MFLOPS during the triangle fill-rate tests

Textured Triangle Fill-Rate

Figure 17b shows an extract of OpenGL pseudo code taken from the textured triangle fill-rate test. Again, it purposely doesn't contain all the code executed during the test but is intended to represent the key elements.

```
.........
glEnable( GL_LIGHT0 );
glShadeModel( GL_SMOOTH );
glEnable( GL_TEXTURE_2D );
.........
glTexParameteri( GL_TEXTURE_2D,
GL_TEXTURE_MAG_FILTER,  GL_LINEAR);
glTexParameteri( GL_TEXTURE_2D,
GL_TEXTURE_MIN_FILTER,
GL_LINEAR_MIPMAP_LINEAR);
glTexParameteri( GL_TEXTURE_2D,
GL_TEXTURE_WRAP_S,  GL_REPEAT);
glTexParameteri( GL_TEXTURE_2D,
GL_TEXTURE_WRAP_T,  GL_REPEAT);
glTexEnvi( GL_TEXTURE_ENV,
GL_TEXTURE_ENV_MODE, GL_MODULATE);
.........
glTexImage2D( GL_TEXTURE_2D, 0, 3, 64, 64, 0,
GL_RGB, GL_UNSIGNED_BYTE,  { });
glTexImage2D( GL_TEXTURE_2D, 1, 3, 32, 32, 0,
GL_RGB, GL_UNSIGNED_BYTE,  { });
glTexImage2D( GL_TEXTURE_2D, 2, 3, 16, 16, 0,
GL_RGB, GL_UNSIGNED_BYTE,  { });
glTexImage2D( GL_TEXTURE_2D, 3, 3, 8, 8, 0,
GL_RGB, GL_UNSIGNED_BYTE,  { });
glTexImage2D( GL_TEXTURE_2D, 4, 3, 4, 4, 0,
GL_RGB, GL_UNSIGNED_BYTE,  { });
glTexImage2D( GL_TEXTURE_2D, 5, 3, 2, 2, 0,
GL_RGB, GL_UNSIGNED_BYTE,  { });
glTexImage2D( GL_TEXTURE_2D, 6, 3, 1, 1, 0,  GL_RGB,
GL_UNSIGNED_BYTE,  { });
.........

START LOOP
glBegin( GL_TRIANGLE_STRIP );
glNormal3fv(-36.3785, -36.3785, -0.695387 );
glTexCoord2fv(-0.695387, 0.181309 );
glVertex3fv(-3.43724, -3.99015, -5.89054 );
glNormal3fv(-36.377, -36.9375, -0.695359 );
glTexCoord2fv(-0.705928, 0.134688 );
glVertex3fv(-3.45947, -4.0145, -5.87967 );
glNormal3fv(-35.8189, -36.604, -0.684835 );
glTexCoord2fv(-0.69964, 0.203727 );
.........
glNormal3fv(-23.338, -8.8734, -0.449494 );
glTexCoord2fv(-0.176747, 0.875623 );
glVertex3fv(-1.45689, -2.77448, -4.9768 );
glNormal3fv(-23.8297, -8.97293, -0.458765 );
glTexCoord2fv(-0.178623, 0.870419 );
glVertex3fv(-1.47405, -2.76471, -4.98318 );
glEnd();
END LOOP
.........
```

Figure 17b, OpenGL pseudo code during textured triangle fill-rate test.

The first three lines show the enabling of gouraud shading, lighting, and 2D texturing.

The next sequence of lines defines the texture state. To understand its meaning fully, it's necessary to have an appreciation of texture mapping and, more specifically, mipmapping. To that end, it is assumed most readers will be familiar with mipmapping; however, for readers who aren't as familiar, a very brief description is given in the following paragraph. Further details are provided in [26]. The GL_MODULATE setting for the texture environment call determines that the final vertex color value will be determined by the texture image and the original vertex color value.

When using texture mapping, the quality of image can be significantly improved using multiple texture images or mipmaps. This is because most vertices after appropriate transformations and scalings will typically lie between four elements on a texture map. These elements are called texels. The final color of the vertex is chosen dependent upon the texture state definitions and can range from using simply the nearest texel to a linear combination of the four surrounding texels. When zooming in and out of geometry, however, even basing the color on the four surrounding texels can result in "jumpiness" in the colors and poor overall visual quality. When using mipmapping a series of textures is defined that reduces in size by powers of two from the original texture image size down to 1x1 pixel. In this case, even after appropriate transformations and scalings, a vertex will typically lie between four texels in two separate mipmaps. The vertex color value can then be determined by some combination of the 8 surrounding texels depending on what the texture state is defined as. This approach yields a

significant improvement in visual quality along with noticeably more realistic overall results.

In the case of the texture triangle fill-rate test quoted above, the mipmaps are precalculated before the test, for obvious reasons, and then explicitly defined by the glTextImage2D calls. Because the original image is 64x64 pixels, 6 additional mipmaps are produced before a 1x1 pixel image results. A total of seven texture images are defined. The GL_MAG_FILTER and GL_MIN_FILTER settings refer to the behavior adopted when a pixel maps to an area smaller than the smallest element in a texture, and when a pixel maps to an area larger than the largest element in a texture respectively. The GL_LINEAR setting for the magnification function dictates that the vertex color value used will be determined by the weighted average of the four texture elements that are closest to it. The GL_LINEAR_ MIPMAP_LINER setting for the minification function results in the final vertex color value being determined by an average from two color values, which in turn are generated from a weighted average from four texels in two mipmaps. The two mipmaps are select so that their size closely matches that of the pixel being generated.

Like the triangle fill-rate test, triangles are sent to graphics as triangle strips, comprising 120 triangles per strip and 10 strips overall. Again, vertices and normals are generated from random numbers; however, in this case texture coordinates, defining the position of the vertex in the texture image, are also sent. As with the vertex and normal values, these are precalculated and stored before the test measurement commences. Clearly, because of the extra computation required to map vertex positions onto texture images combined with mipmapping, the texture triangle fill-rate test involves more computation compared with the pure triangle fill-rate test.

In this case, the effect of downloading textures also influences results. The textures in the test are RGBA, which due to the implementation of O2's MRE do not need to be unpacked by the CPU and therefore use an uncached protocol and do not travel across the SysAD bus. As a consequence, the larger improvement seen with MIPS R12000 relative to QED RM5200 on this test is probably due to the influence of other factors, such as load/store instruction scheduling. Other architectural improvements, such as an increased branch prediction table and active list, combined with other differences relative to QED RM5200, such as out-of-order execution, combine to make texture transfer during the textured triangle fill-rate test yield larger improvements with MIPS R12000.

For both the triangle and textured triangle fill-rate test MIPS R12000 shows less improvement when the triangle size is 50 pixels. The reason for this may not be obvious from the relative results shown in Table 1, above. The relative triangle fill rate results shown in Figure 9, above, show MIPS R12000 to be 25% and 32% faster than QED RM5200 for the triangle and textured triangle fill-rate tests, respectively. The slight drop-off in the 50-pixel triangle and textured triangle fill-rate tests is likely due to the influence of another bottleneck appearing in the graphics pipeline, such as in the rasterization stage. Because the results for the QED RM5200 are not as high in absolute terms, they do not hit this limit. Likewise, because the fill-rate for the textured triangle case is slightly less than that of the non-textured case, the drop-off is reduced.

To investigate how the processor architecture features of MIPS R12000 influence the graphics performance, Table 4 shows graduated instructions/cycle, graduated floating-point instructions/cycle, and graduated load/store instructions per cycle.

	MIPS R10000 250 MHz	MIPS R12000 300 MHz
Graduated instructions per cycle		
1 Pixel	1.220049	1.293745
25 Pixel	1.243763	1.302638
50 Pixel	1.206619	1.311304
Graduated floating-point instructions per cycle		
1 Pixel	0.344053	0.379708
25 Pixel	0.353285	0.386848
50 Pixel	0.341924	0.394801
Graduated load/store instructions per cycle		
1 Pixel	0.458965	0.493239
25 Pixel	0.467848	0.504176
50 Pixel	0.456738	0.498432

Table 4, Graduated instructions per cycle, graduated floating-point instructions per cycle, and graduated load/store instructions per cycle.

Although the textured triangle fill-rate tests may be expected to have additional complexity relative to the triangle fill-rate test because the textures are RGBA format, they can be pulled directly from memory by the MRE, causing minimal overhead to the CPU. As a consequence, although there may be slight additional work to apply textures, the overhead is probably very small. Hence the graduated instructions, graduated floating-point instructions, and graduated load/store instructions per cycle observed during the test would be expected to be very similar to the triangle fill-rate tests with no textures. Clearly, the results in Table 4 show this to be true.

Likewise, the architectural improvements of the MIPS R12000 would also be expected to increase the graduated instructions, graduated floating-point instructions, and graduated load/store instructions per cycle observed during the textured triangle fill-rate test, and again this is shown by the results in Table 4.

Again, because the RGBA textures used in the textured triangle fill-rate test are not unpacked, texturing would not be expected to yield a noticeable increase in traffic between the primary and secondary cache, or between secondary cache and main memory when compared with the untextured triangle fill-rate tests. The results in Table 5 show this to be the case. Likewise, the reasons listed above explaining why the architectural improvements of MIPS R12000 and increasing the graduated instructions per cycle would typically yield to improvements in L1-L2 and L2-main memory bandwidth as well as the observed MFLOPS also apply.

Table 5, The L1-L2 bandwidth used, memory bandwidth used, and average MFLOPS seen during the test.

	MIPS R10000 250 MHz	MIPS R12000 300 MHz
L1-L2 bandwidth		
1 Pixel	12.94796	16.53812
25 Pixel	14.36574	18.06257
50 Pixel	17.85962	22.61384
Memory bandwidth used		
1 Pixel	1.622079	2.04869
25 Pixel	1.689756	2.108009
50 Pixel	1.805317	2.227033
Average MFLOPS		
1 Pixel	88.4582	106.545
25 Pixel	87.57669	111.4861
50 Pixel	99.86854	118.4403

Texture Upload

Figure 17c shows an extract of OpenGL pseudo code taken from the texture upload test, again purposely only showing key elements.

```
.....
glTexParameteri( GL_TEXTURE_2D,
GL_TEXTURE_MIN_FILTER,
GL_LINEAR_MIPMAP_LINEAR);
glTexParameteri( GL_TEXTURE_2D,
GL_TEXTURE_MAG_FILTER, GL_NEAREST);
glTexEnvi( GL_TEXTURE_ENV,
GL_TEXTURE_ENV_MODE, GL_DECAL);
glTexImage2D( GL_TEXTURE_2D, 0, 3, 64, 64, 0,
GL_RGB, GL_UNSIGNED_BYTE,  { });
glTexImage2D( GL_TEXTURE_2D, 1, 3, 32, 32, 0,
GL_RGB, GL_UNSIGNED_BYTE,  { });
glTexImage2D( GL_TEXTURE_2D, 2, 3, 16, 16, 0,
GL_RGB, GL_UNSIGNED_BYTE,  { });
glTexImage2D( GL_TEXTURE_2D, 3, 3, 8, 8, 0,
GL_RGB, GL_UNSIGNED_BYTE,  { });
glTexImage2D( GL_TEXTURE_2D, 4, 3, 4, 4, 0,
GL_RGB, GL_UNSIGNED_BYTE,  { });
glTexImage2D( GL_TEXTURE_2D, 5, 3, 2, 2, 0,
GL_RGB, GL_UNSIGNED_BYTE,  { });
glTexImage2D( GL_TEXTURE_2D, 6, 3, 1, 1, 0,  GL_RGB,
GL_UNSIGNED_BYTE,  { });
.....
glBegin( GL_TRIANGLE_STRIP );
glTexCoord4f(0.5, 0.5, 0, 1);
glVertex3f(0, 0, -1);
glTexCoord4f(0.515625, 0.5, 0, 1);
glVertex3f(0.00520833, 0, -1);
glTexCoord4f(0.5, 0.515625, 0, 1);
glVertex3f(0, 0.00520833, -1);
glEnd();
glFinish();
.....
glAreTexturesResidentEXT(2, 0x107308d0, 0x107d1190);
.....
BEGIN LOOP
glBindTextureEXT( GL_TEXTURE_2D, 1);
glBindTextureEXT( GL_TEXTURE_2D, 2);
END LOOP
.....
```

Figure 17c, OpenGL pseudo code during texture upload tests.

As, with the texture triangle test, the first few lines set the magnification and minification filters and the texture mode, and define the mipmaps. The next section of code draws a triangle strip; however, since the purpose of the test is to measure texture upload performance, this is only to force the textures to be applied to geometry. During the test itself, the transfer of textures from system to texture memory is forced by alternating calls to GLBindTexture where the target, or texture, is bound to alternating mipmaps. This causes the two textures to be uploaded continually during the course of the test. The call to glAreTexturesresidentEXT before the test begins queries whether the textures are resident in local texture memory, or main memory. This is to verify that indeed the textures will be uploaded during the test and not stored within local texture memory. Clearly on O2 texture memory is part of main memory.

The results in Table 1 show that the effect of unpacking the RGB texture on QED RM5200 is minimal, whereas on MIPS R12000 the effect is more noticeable. As mentioned above, unpacking the texture forces an extra return trip between the CPU and MRE/memory. It is likely that on MIPS R12000 this may be yielding some interference in secondary cache and possibly congestion on the system bus, which lead to a reduction in the performance. Like the results comparing triangle and textured triangle fill-rate performance shown in Figure 9, it should be remembered that MIPS R12000 demonstrates noticeably better texture upload performance relative to QED RM5200 and thus any effects arising through congestion or other such bottlenecks may appear on these systems. Since primitive level performance is specifically designed to test the limit of a machine in specific areas, application performance will probably not be subject to the results of these specific effects due to the influence of other non-graphics code and instructions during graphics activities.

	MIPS R10000 250 MHz	MIPS R12000 300 MHz
Graduated instructions per cycle		
RGB	1.126785	1.159997
RGBA	1.140506	1.205605
Graduated floating-point instructions per cycle		
RGB	0.027162	0.029528
RGBA	0.028074	0.029657
Graduated load/store instructions per cycle		
RGB	0.398614	0.425494
RGBA	0.407575	0.420658

Table 6, Graduated instructions per cycle, graduated floating-point instructions per cycle and graduated load/store instructions per cycle.

The main purpose of the texture upload test is to measure the speed with which textures can be transferred from memory to the graphics hardware. When compared to rendering geometry, the number of graduated instructions may be expected to reduce. This is because there is not the same number of calculations performed since there are fewer geometry transformations. The results in Table 6 concur with this reasoning, especially when compared with the triangle and textured triangle fill-rate tests shown in Tables 2 and 4, respectively.

Likewise, since the RGB textures have to be unpacked by the CPU, this would be expected to yield an increase in graduated instructions but not graduated floating-point instructions per cycle. In fact, since there are very few geometry calculations, the graduated floating-point instructions would be expected to be significantly less in the texture upload case compared with the triangle and textured triangle fill-rate tests. Such a result would not be expected for the load/store instructions since the reduced number of load/stores generated by the fewer geometry calculations will be potentially offset by the increased number of load/stores arising from the texture uploading.

	MIPS R10000 250 MHz	MIPS R12000 300 MHz
L1-L2 bandwidth		
RGB	15.2654	19.54628
RGBA	7.043158	8.215042
Memory bandwidth used		
RGB	2.303807	2.087869
RGBA	1.823367	2.044901
Average MFLOPS		
RGB	6.790501	8.858251
RGBA	7.018386	8.897163

Table 7, The L1-L2 bandwidth used, memory bandwidth used, and average MFLOPS seen during the test.

Again like the previous tests, these predictions concur with the observation given in Table 6. As a consequence of RGB textures being unpacked by the CPU, whereas RGBA are not, the bandwidth between cache and main memory may be expected to increase. That is, unless the cache is being very efficient, which would yield an increase between the primary and secondary cache levels but not necessarily between secondary cache and main memory. Clearly, the results in Table 7 show how well the secondary cache is working when unpacking textures for the RGBA case. It's also interesting to note that the L1-L2 bandwidth is significantly less than in both the triangle and textured triangle fill-rate tests and is likely due to the significant reduction in geometry calculations during the texture upload test.

The overall memory bandwidth observed is very similar to both the triangle and textured triangle cases and shows that the secondary cache line reuse for all tests is high. Requests for data, therefore, are typically satisfied from secondary cache without requiring loads from main memory. To illustrate this point further, Table 8 compares the secondary cache line reuse seen during the texture upload test for both the RGB and RGBA cases. The results clearly show that where there is more CPU activity the cache line reuse increases significantly.

	MIPS R10000 250 MHz	MIPS R12000 300 MHz
L2 Cache line reuse		
RGB	24.326231	33.894377
RGBA	7.018386	14.161825

Table 8, Secondary cache line reuse observed during texture upload test.

The significant reduction in floating-point activity arising from the relatively low number of geometry calculations in the texture upload test is the main reason why the observed MFLOPS shown in Table 8 are dramatically less than those seen in the triangle and textured triangle fill-rate tests. Because the RGB and the RGBA texture upload tests have the same amount of geometry calculations, the observed MFLOPS for both would be fairly similar, which is also demonstrated by the results.

This section has demonstrated how the combination of QED RM5200 and MIPS R12000 microprocessors complement O2's architecture to yield a synergistic effect on low-level graphics performance. By itself the fact that many of the improvements seen in graphics primitive performance are significantly above clock-rate increases, when compared to the previous microprocessors in both families, is impressive. Compared to the increases seen in pure CPU performance, as shown by the SPECint95 and SPECfp95 results, it is a very significant achievement and demonstrates well the strengths of O2's architecture. The next section shows how both the CPU and graphics improvements translate to the benefits seen by an end-user.

Translation of Improvements Afforded through the New CPUs into End-User Benefits

The low-level graphics primitive results very clearly demonstrate the performance synergy arising from O2's CPU and architecture. This alone, however, does not truly represent what an end-user will see. Although it is very clearly dependent on a specific application or code segment, potentially the improvement in application performance can be diluted slightly. To demonstrate the improvements afforded through the O2's new CPUs, as seen by an end-user, it is necessary to show the effect they have on an example application. To this end, this section uses two applications developed by CADCENTRE, Review Reality and PDMS. Review Reality allows users to visualize structural models and assemblies, such as oil platforms, with a very high degree of realism, taking advantage of texture mapping. PDMS is aimed at a slightly previous stage to this in the workflow where such models are created and assembled.

A series of tests that were developed as being representative of key aspects of users workflow was performed for both Review Reality and PDMS. For Review Reality there were two series of tests. One comprised timing a single repaint, i.e., the time taken to draw the model and realize it to the screen. This was performed for several models at several different levels of accuracy of representation, i.e., how round curves appeared in the final display. The second series of tests comprised timing the playback of a predefined animation path around a model, again for several levels of accuracy. Since Review Reality is primarily focused at visualization of models, being able to draw models fast with high accuracy is of a key concern to users. To that end, the repaint tests demonstrate a system's ability to meet users' needs. Likewise, the animation playback test demonstrates a system's ability for a user to traverse and manipulate a model in real time and again represents a key element in Review Reality's workflow. Figure 18 shows an example image from the Review Reality program.

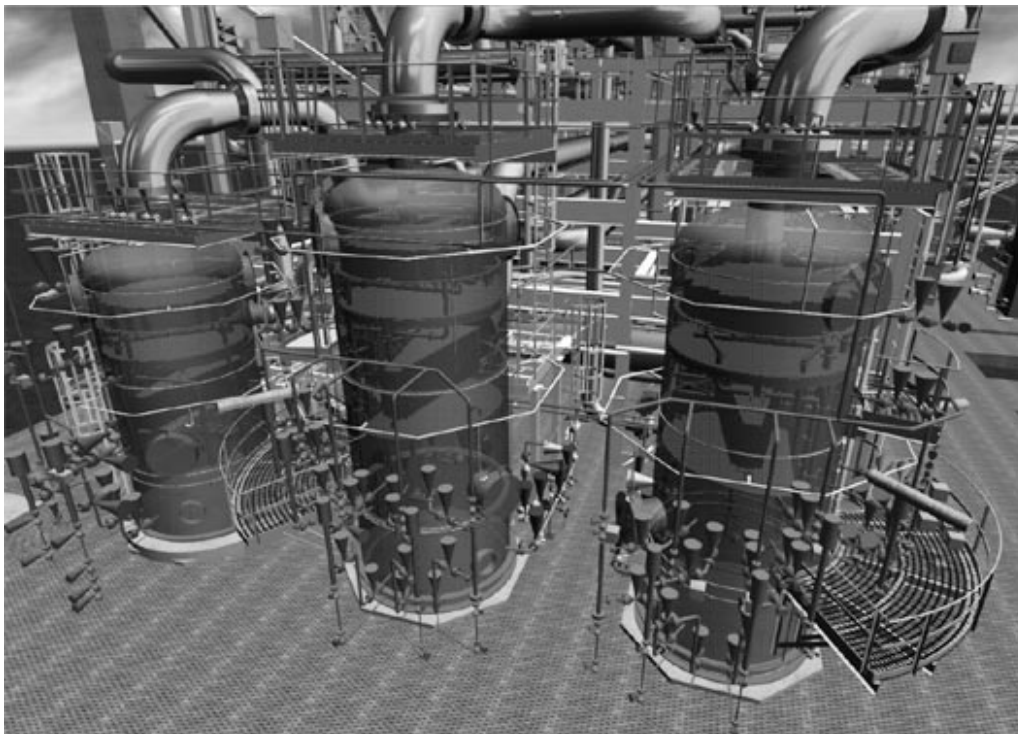


Figure 18, Sample image from Review Reality program.

Similar to the Review Reality tests, the PDMS tests focus on a user's workflow within this application. There are four tests focused on individual tasks that are considered to be key elements in a user's typical work. They are: adding members within a structure, zooming in/out of a model, rotating a model, and creating a series of additional views. It should be pointed out that the PDMS tests were performed using a previous version of PDMS, version 5.1.1. Although to the user this visually doesn't appear too dissimilar to the latest version, except of course it doesn't contain the improvements afforded by the latest version, it is fairly different at a system level since it is based on IrisGL and not OpenGL. Since IrisGL was the forerunner to OpenGL, it has a lot of similarities; however, it is also different in many fundamental respects and implementations of the code. Silicon Graphics firmly maintains binary compatibility between IrisGL and OpenGL; however, since all current systems are designed with OpenGL in mind, IrisGL applications execute using a translator, translating IrisGL calls to OpenGL (affectionately called IGLOO within SGI). This results in a greater amount of system activity when running such IrisGL programs and will clearly influence improvements seen compared with native OpenGL applications. The latest version of PDMS is OpenGL based.

For more information on both Review Reality and PDMS as well as CADCENTRE in general see [27].

		Performance of QED RM5200 Compared with MIPS R5000	Performance of MIPS RI2000 Compared with MIPS RI0000
PDMS test suite			
Add Members		1.36x	1.21x
Zoom In/Out		1.33x	1.17x
View Rotate		1.34x	1.15x
Add Views		1.18x	1.07x
Review Reality test suite Time for one screen repaint			
Costain model	Wireframe	1.29x	1.09x
Costain model	Shaded - low	1.31x	1.13x
Costain model	Shaded - medium	1.35x	1.16x
Costain model	Shaded - high	1.38x	1.18x
Costain model	Shaded - extreme	1.39x	1.19x
Time for one screen repaint			
Dunbar model	Wireframe	1.32x	1.10x
Dunbar model	Shaded - low	1.35x	1.14x
Dunbar model	Shaded - medium	1.37x	1.16x
Dunbar model	Shaded - high	1.41x	1.18x
Dunbar model	Shaded - extreme	1.44x	1.19x
Animation playback			
Brent model	Wireframe	1.26x	1.09x
Brent model	Shaded	1.21x	1.08x
Brent model	Shaded & textures	1.16x	1.06x
Clock speed		1.50x	1.20x
SPECint95		1.49x	1.20x
SPECfp95		1.16x	1.07x

Table 9, Improvements in CADCENTRE PDMS and Review Reality tests on QED RM5200– and MIPS RI2000–based systems.

When looking at the improvements of QED RM5200 and MIPS RI2000 on PDMS performance, the effect of the IGLOO graphics layer becomes noticeable, especially when these improvements are compared with those of the graphics primitive results shown in Table 1. The PDMS "Add Views" test also incurs a lot of system overhead due to the creation of new windows. In both cases, the net effect will be to reduce the effect of raw graphics performance improvements. The fact that the improvements for the PDMS tests lie between the SPECint95 and SPECfp95 results highlights this and demonstrates the increased influence of CPU activities in application performance compared with graphics primitive performance. Since the MIPS RI2000 results lie closer to the SPECfp95 increase, whereas the QED RM5200 results lie in the middle of the SPECint95 and SPECfp95 increases, this would suggest that features such as increased branch prediction and increased MRU table are possibly yielding some noticeable effects.

Again, the repaint results for Review Reality, although OpenGL based, show an influence arising from extra CPU activity during the test, such as traversal of internal data structures, view frustum culling, and so on. Hence, these results are slightly reduced from those of the graphics primitives given above. It's very interesting to note, though, that in both cases as the model complexity increases, such as with the high and extreme cases, both QED RM5200– and MIPS RI2000–based systems improve their respective performances. This is likely to be due to the fact that the benefits of O2's architecture show through when graphics calculations become a more dominant activity compared with the application CPU activity. It should also be noted that the graphics primitive performance results above implied that the MIPS RI2000–based O2 system exhibited some fill limitations for larger triangles and hence this may also be subduing the performance of the MIPS RI2000–based O2 system during the Review Reality tests.

For the animation Review Reality test case, the additional CPU processing arising from traversing the animation path as well as associated calculations is slightly further reducing the improvement seen compared with the pure graphics performance improvements. Since the animation path travels very close to certain components within the model, it is likely that both QED RM5200– and MIPS RI2000–based systems are showing some fill limitations, the net effect of which will be to reduce slightly the overall performance improvements. It's interesting to note that as the model complexity rises and textures are included, the improvements seen by MIPS RI2000 over MIPS RI0000 remain relatively constant. For the QED RM5200 compared with MIPS R5000, the improvement noticeably drops when textures are included, likely because of the architectural differences of MIPS RI2000 compared with QED RM 5200. Differences such as four instruction decode and issue per cycle and out of order execution, as detailed above, are likely to allow the processor to deal with the additional load arising through increased model complexity and texturing.

In summary, the above results show that the improvements yielded through the new CPUs are typically closer to the upper end of the spectrum one would expect to see, as indicated by the SPECint95 results. Although the performance improvements seen in the application are lower than those seen in low-level graphics primitives performance improvements, this is to be expected when taking into consideration the other factors involved. Overall the results are a testimonial to O2's architecture, which also yields demonstrable benefits with new CPUs. The examples provided in this section show these improvements are of direct benefit to an end-user in terms of increased productivity on key tasks within typical workflow.

Conclusion – Resuming Play

O2's architecture is unique in the low-cost UNIX marketplace and affords significant benefits to an end-user. The lst and l0 system described in the introduction is a very good example of this, and the previous sections in this paper have also described others. Likewise, because O2's architecture is unique, it is often easy to assume improvements in some areas will only yield benefits in those areas and overlook significant improvements in other areas. Such assumptions would be untrue, and this paper has demonstrated how CPU improvements also yield significant benefits to O2's graphics performance. Without tempering such results with the realism of what a user sees, i.e., application performance, an over-optimistic picture of benefits may be achieved. To this end, this paper has also demonstrated how both CPU and graphics primitive performance improvements translate into the benefits seen by an end-user running an application, in this case CADCENTRE's Review Reality and PDMS.

When considering O2's performance relative to competitive systems it would appear at first sight that the competitive systems are strong. Such comparisons don't convey the whole story, however, since they don't emphasize O2's strengths, which lie in its system architecture and its features. The SPEC results relative to Sun's new processors, the UltraSPARC II 360 MHz and UltraSPARC IIi 333 MHz, show an apparent strength for Sun's platforms. Unfortunately, though the real world is rarely so well behaved, and while Sun may appear to have a strong position in CPU performance, that strength changes when taking into consideration other factors, such as the price performance of the complete system to perform texturing and video. Indeed, very few if any standard current competitive systems are able to perform such sophisticated digital media capabilities offered in a base level O2.

It should also not be forgotten that the apparent strength of many microprocessors on the SPECcpu95 tests is often due to significant optimization for these specific tests. Such optimization is very impractical when considering real applications, since the amount of code is too great to allow such dedicated tuning efforts for all tasks that users perform. To demonstrate what effect this can have, Figures l9a and l9b compare the ratio of SPECint95 to SPECint_base95 and SPECfp95 to SPECfp_base95 for MIPS RI2000, UltraSPARC II, and UltraSPARC Iii.

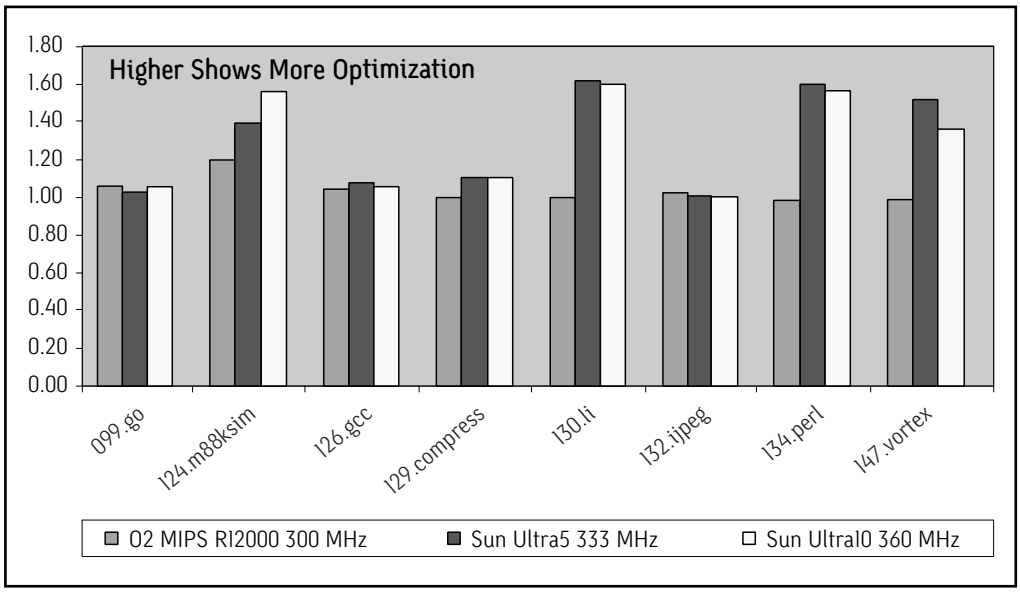


Figure 19a, Comparison of SPECint_base95 to SPECint95 for O2 R10000 and Sun Ultra5 and Ultra10 systems.

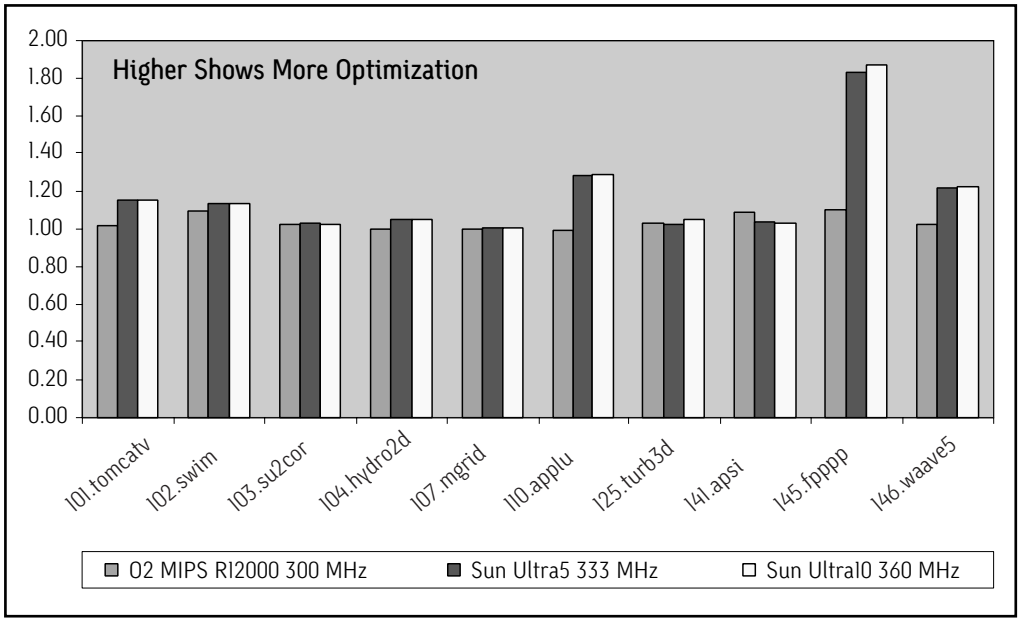


Figure 19b, Comparison of SPECfp_base95 to SPECfp95 for O2 R10000 and Sun Ultra5 and Ultra10 systems.

As can be seen from Figures 19a and 19b, both Sun microprocessors benefit significantly from the large optimizations of one or two tests out of the entire suite. Since such benefits are significantly more difficult to realize at application level, there will be many situations where even with apparent strong competition, both O2 QED RM5200 and O2 MIPS R12000 systems will be very competitive. This is demonstrated by the results for the SPECfp95 146.wave5 test shown in Figure 11 in the section entitled Computation Performance and Overall System Throughput. The influence of software optimization on this test for MIPS R12000 is minimal, as shown in Figure 19b; however, it is very significantly dominant compared to both UltraSPARC II 360 MHz and UltraSPARC IIi 333 MHz. Since the 145.fpppp test is taken from a real application, it very clearly shows that the real world is often somewhat different from what would be expected.

As shown in the sections above, the performance improvements afforded by QED RM5200 and MIPS R12000 are very significant for low-level graphics primitives. Even when translated into the improvements seen with applications, such as CADCENTRE Review Reality and PDMS, they are still very significant and clearly demonstrate benefits to the end-user. These comparisons, however, don't perhaps convey the true story in terms of benefits for the end-user since they only compare relative to the last generation of processors, including MIPS R5000 200 MHz and MIPS R10000 250 MHz. In reality, most O2 customers are probably using a previous generation of these two microprocessors, which means that the benefits for those customers will be significantly higher still. Figure 20 shows the improvements in SPECint95, SPECfp95, and triangle fill-rates for O2 systems using QED RM5200 and MIPS R12000 relative to all existing O2 configurations.

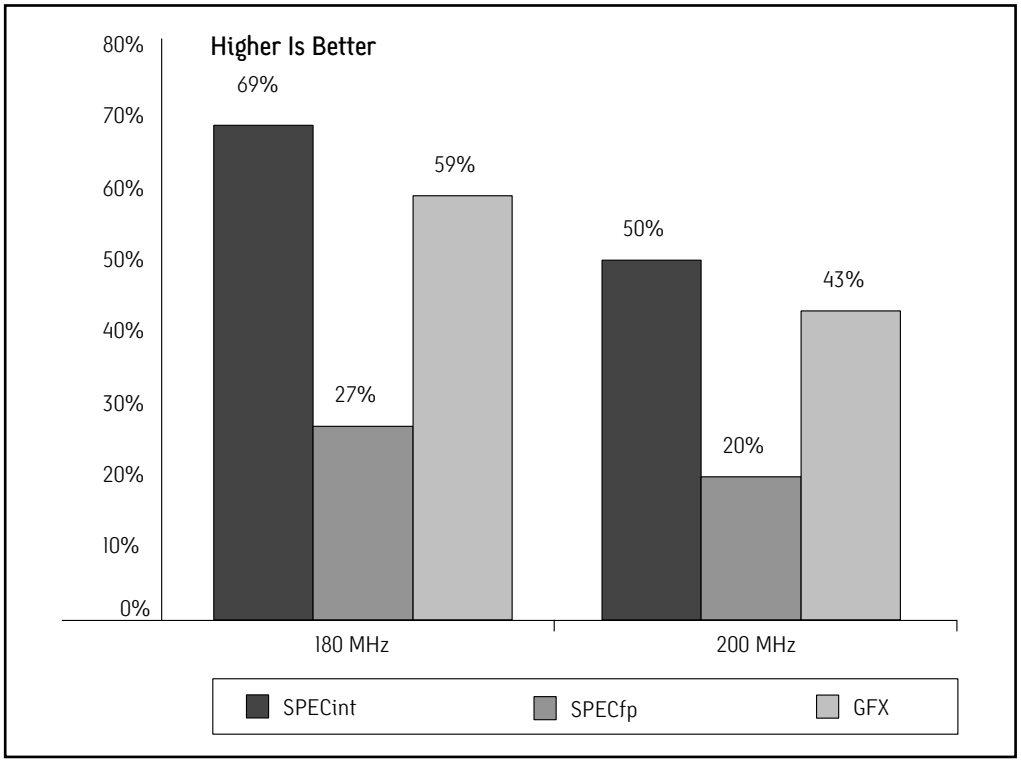


Figure 20a, Improvements afforded through QED RM5200 300 MHz native to existing MIPS R5000 systems.

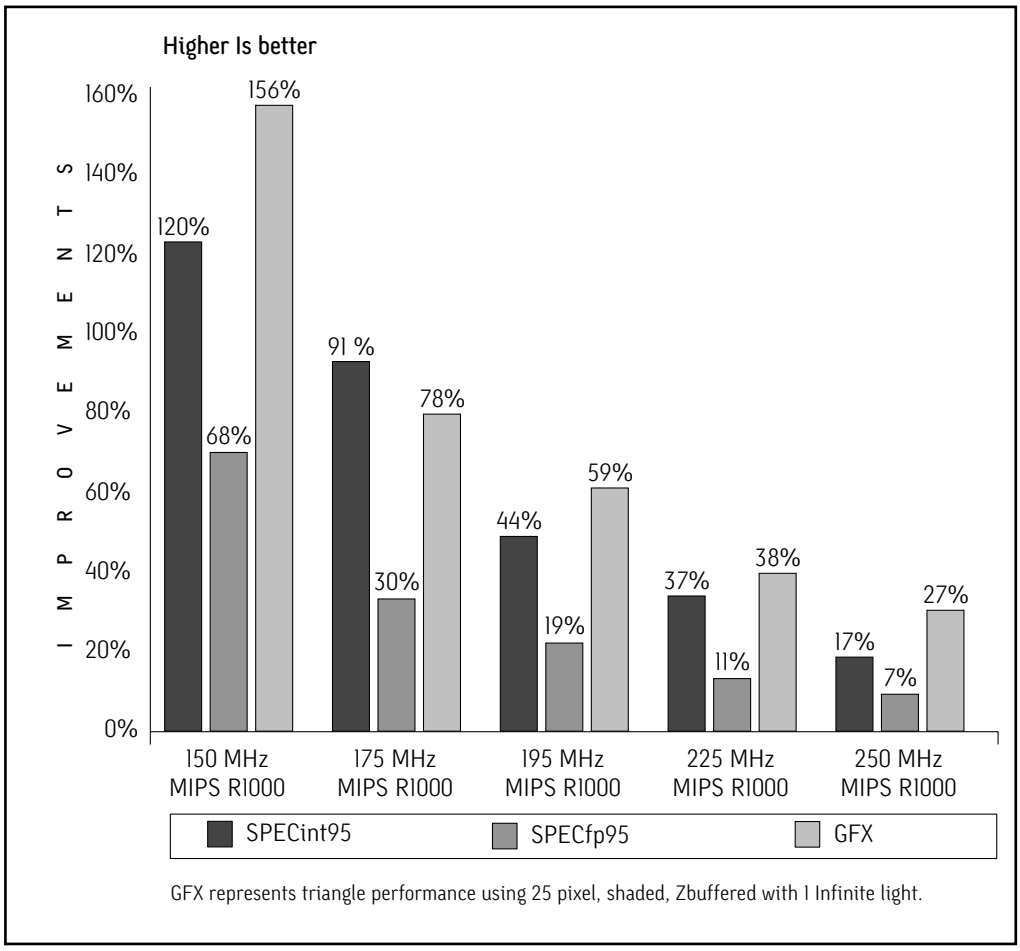


Figure 20b, Improvements afforded through MIPS R12000 300 MHz native to existing MIPS R1000 systems.

Figures 20a and 20b clearly show that for many customers the benefits of QED RM5200 and MIPS R12000 are well over 100%. This is very significant considering that to realize such benefits one simply has to change the processor module.

Moving forward there will continue to be further CPU developments to O2 in both the MIPS R12000 and QED RM5200 microprocessor families. As has been shown in this paper, these will yield significant benefits to the end-user in both graphics primitive level and, more importantly, in application performance. Combined with the strengths of O2's unique architecture and the sophisticated digital media features it affords, these improvements will continue to ensure that O2 enables users to improve productivity and gain significant competitive advantage. SporTVision's 1st and 10 system is clearly a good and very real example of this.

Acknowledgments

As ever, the results of one are dependent on the efforts of many, and to that end I would like to thank the following people for their help in producing this paper: John Schimpf, Jeff Brown, Ujesh Desai, Bob Sanders, Dale Ah Tye, Ken Yeager, Andrew Walton, Gianpaolo Tommassi, all of Silicon Graphics; Alan Clegg of CADCENTRE (UK); and Andy Keane, Tom Hampton, and Van lewing of QED inc.

References

1. SportVision systems, www.sportvision.com/products_first+ten.htm.
2. "The '1st and 10' line makes its mark," www.usa.cannon.com/feature/dec98feature/index.htm.
3. "O2 technical Report version 1.0," Silicon Graphics, October 1997.
4. "PCI user manual."
5. "AGP Specification," Intel Corporation.
6. "The Ultra5 and Ultra10 Workstation Architecture—Technical White paper," Sun Microsystems, January 1998.
7. "UPA—Sun's High-Performance Graphics Connection—Technical White paper," Sun Microsystems, 1999.
8. "Sun Graphics Benchmarks," Sun Microsystems, 10/28/1998.
9. OpenGL Performance Characterization Committee, part of the Standard Performance Evaluation Corporation, www.spec.org.—*put actual viewperf result web address here*.
10. "UltraSPARC and New-Media Support," Sun Microsystems, Microelectronics.
11. "UltraSPARC New-Media Processing: New Tasks, New Processors, New System Architectures," Sun Mircosystems, Microelectronics.
12. "UltraSPARC™ Turbocharges Network Operations and New Media Computing," Sun Microsystems, Microelectronics.
13. "UltraSPARC™ Ultra Port Architecture (UPA): The New-Media System Architecture," Sun Microsystems, Microelectronics.
14. "O2 Digital Media White Paper," Silicon Graphics 1999 - put web address here.
15. SPEC CPU95 test results, Standard Performance and Evaluation Committee, www.spec.org.
16. "Intel Pentium III Processor Performance Brief," Intel corporation, http://www.intel.com/procs/perf/PentiumIII/brief/tables.htm.
17. LUMIS Shake benchmark results, see http://www.lumis.com/tests/shake/index.shtml.
18. LUMIS Maya Rendering benchmark results, see http://www.lumis.com/tests/maya/render.shtml.
19. "High-Performance Computer Architecture," 3rd Edition. Harold S. Stone, Addison-Wesley, 1993.
20. "Computer Architecture Single and Parallel Systems," Mehdi R. Zargham, Prentice Hall, 1996.
21. "An Illustration of the Benefits of MIPS R12000 and OCTANE System Architecture, Ian Williams, Silicon Graphics, 1999. http://www.sgi.com/octane/images/octane.pdf.
22. "MIPS R5000 Microprocessor Technical Backgrounder," MIPS Technologies, www.mips.com.
23. "QED RISCMark RM5271 64-Bit Superscalar Microprocessor," Quantum Effect Design, Inc.
24. "The MIPS R10000 Superscalar Microprocessor," Kenneth C. Yeager, IEEE Micro, April 1996.
25. "200-MHz Superscalar RISC Microprocessor," Nader Vassenghi, Kenneth Yeager, Egino Sarto, and Mahdi Seddighnezhad, IEEE Journal of Solid-State Circuits, Vol. 31, No. 11, November 1996.
26. "OpenGL Programming Guide," 2nd Edition, Addison-Wesley, ISBN 0201461382.
27. CADCENTRE Group, www.cadcentre.com.

Appendix I — SporTVision's 1st & Ten System

The following information describing its 1st & Ten system was kindly supplied by SporTVision.

About SporTVision

SporTVision, which was launched in January 1998, develops technology-based enhancements for sports television, the Internet, and new media platforms. SporTVision's engineering team developed the 1st & Ten system, which creates and displays a 1st down line in football games. Launched on ESPN this season, 1st & Ten was one of the most popular new sports innovations in years. SporTVision's first system, AIRf/x, measured the vertical leap of basketball players and was seen on TNT and Turner Broadcasting. The same technology (called "MaXAir") was used to measure the altitude gained by the athletes in the half-pipe events on ESPN's "X Games" and in various events at ESPN's "Winter X Games." SporTVision also serves as broadcast technology consultant to the National Football League, the National Hockey League, and the New York Mets. SporTVision's headquarters are in New York, and its development facility is in Mountain View, California.

Overview

This is a brief summary of the technical nature of SporTVision's 1st & Ten system. 1st & Ten enables television football fans to see the 1st down line as easily and clearly as they see the goal line or any 5- or 10-yard line. It does so by painting a virtual 1st down yard line in video, and this line appears to be on the field under foot, just like the actual yard lines. The only visible difference is that the 1st down line is yellow (or any other color the producer chooses).

How It Works

The Silicon Graphics O2 visual workstation is the central computer in the 1st & Ten system that examines every frame of video in real time (i.e., 30 times per second) and determines which pixels to change to yellow. These are all the points in the image, where an actual painted-on-the-field 1st down line would be visible, such as grass along the line that is not obscured by a player or referee. It determines which pixels to change based on very precise information about the camera's view, a 3D model of the field, which camera is on air, and a palette of colors for the field and another palette for players. Pixels along the line with colors from the field palette are changed to yellow unless that color is also in the palette for players. Player colors and other colors not on the field palette are left unchanged. This makes the virtual line visible, where the field is visible, and hidden where the field is obscured, just as a real line would be.

Each camera in the 1st & Ten system is equipped with very precise encoders for pan, tilt, zoom, focus, and extender (1x or 2x doubler). A computer at each camera reads the encoders and transmits these readings to the SportVision production truck 30 times per second. Another computer in the truck gathers readings from all the cameras and transmits a consolidated data stream to the central computer. These readings and the 3D field model go into a geometrical calculation on the O2 that determines which pixels in the video frame would be in an unobstructed view of a real 1st down line.

Yet another O2 workstation determines (also 30 times per second), which camera is tallied (on air). It does this by comparing the video streams from each of the 1st & Ten cameras to the program video. This computer allows for graphics, such as the constant time and score box, that are not in the camera view, to be introduced into the program video. The resultæcamera one, two, three, or none of them, is transmitted to the same computer that is consolidating data from the three cameras, and it adds tally to the data stream going to the central computer.

The final Silicon Graphics IMPACT workstation (soon to be replaced with an O2), has only one simple but crucial task: draw the yellow line in video 60 times per second (every field, not just frame) and send that to a linear keyer to superimpose the yellow onto the program video.

So there are seven computers (three SGIs and four PCs), three sets of special encoders, and abundant wiring dedicated to generating the virtual 1st down line in video format. The data collection and computation requires time, and the virtual 1st down line must be superimposed on the program video at exactly the correct field every 60th of a second, requiring substantial video and audio processing.

Program feed without the 1st & Ten line comes from the primary production truck into SportVision's 1st & Ten truck. Digital feeds from each of the three game cameras also come from main production to 1st & Ten. Program video then goes (undelayed) to the central computer and to a series of frame delays. The camera and program feeds are combined by a quad split unit into a single video feed that goes (undelayed) to the tally computer, which determines which of the three cameras (if any) is on air, as described above. The audio feed goes to an audio delay to be synchronized with the delayed video. To synchronize the computed virtual 1st down line with the program feed, the 1st & Ten system delays program video and audio a fixed number of frames. The delayed program video and the generated virtual 1st down line are both input to a linear keyer that combines the two and outputs the video that will go to air.

Setup

The precision of the 1st and Ten system requires careful setup, accurate surveying of the field and camera locations, and exact timing calibration. Setup routinely takes a day. Football fields usually have crownæ The field is raised several inches in the center, to shed water. The virtual 1st down line must follow the contour of the field, just as the real lines do, so we measure the field elevation at numerous points using a laser plane. Similarly, the camera position, zoom, and focus are critical elements in the geometrical calculations done 30 times per second. So we physically secure the cameras and, using the camera itself, measure its position in 3-dimensional space and the effects of zoom, focus, extender, and lens distortion.

Data communications requires cables from each camera location to the 1st and Ten production truck. Setup usually requires running cables and ringing them out. Synchronizing the calculation with the delayed video is essential. Errors in timing would produce bizarre effects, such as the virtual 1st down line moving ahead of a camera panning motion. This synchronization is done in the central and final computers.

Operation

1st and Ten must not interfere with normal production, so the system architecture and operation are specifically designed to integrate smoothly with standard operations. 1st and Ten operates in a separate 50-foot video production truck, connected to the main production truck. The 1st and Ten operators and engineers work in the SportVision truck and communicate with the producer and director over the intercom. One operator is responsible for correctly placing the 1st down line and for adding or removing it from the program at appropriate times. That operator has a colleague in the stands calling out the official 1st down line position and communicating by intercom. The other operator is responsible for setting and maintaining the field and player color palettes. These colors may change during the course of a game due to lighting changes (say, from the sun setting or rain falling), the field being torn up, or video operations changes.

Corporate Office
1600 Amphitheatre Pkwy.
Mountain View, CA 94043
(650) 960-1980
www.sgi.com

U.S. 1(800) 800-7441
Europe (44) 118-925.75.00
Asia Pacific (81) 3-54.88.18.11
Latin America 1(650) 933.46.37

Canada 1(905) 625-4747
Australia/New Zealand (61) 2.9879.95.00
SAARC/India (91) 11.621.13.55
Sub-Saharan Africa (27) 11.884.41.47

2478 8/99

J10790