To our customers,

Old Company Name in Catalogs and Other Documents

On April 1st, 2010, NEC Electronics Corporation merged with Renesas Technology Corporation, and Renesas Electronics Corporation took over all the business of both companies. Therefore, although the old company name remains in this document, it is a valid Renesas Electronics document. We appreciate your understanding.

Renesas Electronics website: http://www.renesas.com

April 1st, 2010
Renesas Electronics Corporation

Issued by: Renesas Electronics Corporation (http://www.renesas.com)

Send any inquiries to http://www.renesas.com/inquiry.

Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
5. When exporting the products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations. You should not use Renesas Electronics products or the technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations.
6. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
7. Renesas Electronics products are classified according to the following three quality grades: "Standard", "High Quality", and "Specific". The recommended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below. You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application categorized as "Specific" without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for an application categorized as "Specific" or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is "Standard" unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.

   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots.

   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; safety equipment; and medical equipment not specifically designed for life support.

   "Specific": Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.
8. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

(Note 1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

(Note 2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

RENESAS

Phase-out/Discontinued

3D Graphics Accelerator

POWER VR™ Architecture

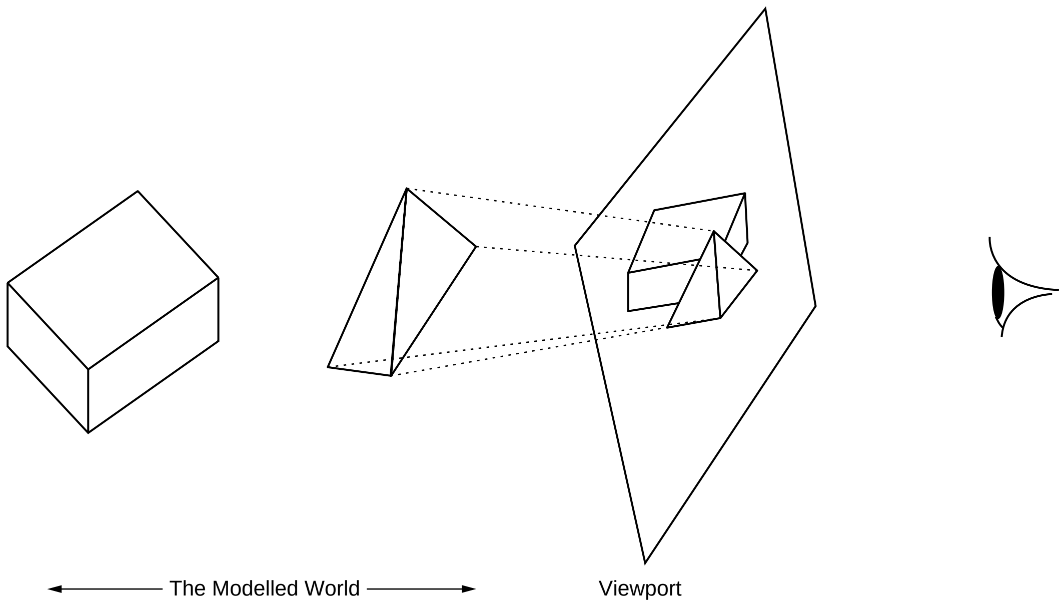
Demystifying Three-Dimensional Graphics Technology

Video games have been around for several years and improve substantially with each generation. Until recently, most of the video game machines and their associated games titles have been restricted by what is known as two-dimensional graphics. The images that such machines or games provide are generally “flat” and lack any notion of depth (*z*-dimension). For example, in a two-dimensional game, the player can only move in *x* and *y* dimensions and cannot look “behind” objects in a scene. In a machine capable of processing three-dimensional graphics, the player has total freedom of movement and can go “around” and “behind” objects, much like in real life.

The processing power and graphics capabilities needed to migrate from two-dimensional to three-dimensional graphics is substantial. In a two-dimensional game, two-dimensional images called *bitmaps* are moved from one area of memory (the source) to another (the destination). Such bitmaps may be combined with other bitmaps or with the destination bitmap to create special effects. Bitmap moves, often referred to as *bitblit*, can also be made conditional on a pixel-by-pixel basis, depending on the destination and source pixels and their relative values.

In a three-dimensional environment, the entire database representing the modeled “world” is processed for every frame displayed. An observer can be looking at and manipulating the world from a new position and direction, but the relative position and orientation of each object is recalculated in every frame. This operation is called *geometry transformation*.

Figure 1. The Basics of Three-Dimensional Processing

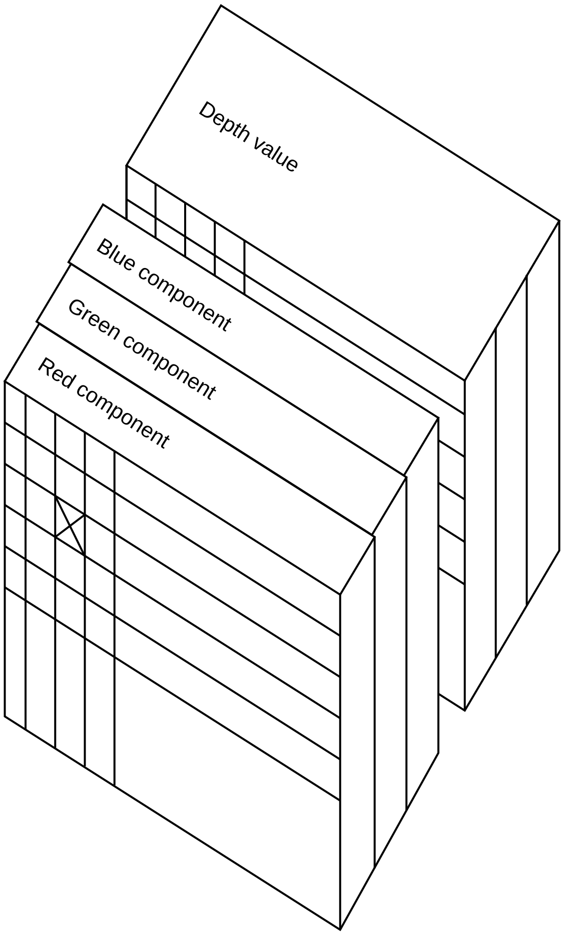


Once the new coordinate of every object is known, the effective view as seen by the observer must be generated, often by projecting the three-dimensional world onto a viewport that constitutes the image seen by the observer. To produce the correct image, it must be known which object or part of each object is visible in the viewport. This operation is generally known as *hidden surface removal*.

Conventional Hidden-Surface Removal Techniques

Many three-dimensional rendering techniques use a z-buffer concept, whereby objects used to construct the three-dimensional world are normally composed of triangles, or *polygons*. For example, the pyramid (tetrahedron) in Figure 1 consists of four polygons. To simplify the rendering hardware, particularly for shading and texturing, many systems use triangles as the polygon primitives. The cube as shown in Figure 1 can be divided into 12 triangles by separating each quad side into two triangles. Each polygon is then drawn, pixel by pixel, into the display buffer, and then displayed on the screen.

Figure 2. Representation of Z Buffer



During this process, polygons or portions of polygons drawn on to the screen are overwritten if follow-on polygons are closer to the observer. Similarly, if polygons or a portion of the polygons to be drawn are behind previously drawn polygons, the associated pixels are ignored and not drawn. To achieve hidden-surface removal, many applications store a z value in addition to a color value for each pixel on the screen. This z value represents depth and is used to determine whether pixels from a newly drawn polygon are in front of or behind the current pixel. Figure 2 shows how each pixel is expressed in red, green and blue components and also in a z value stored in the z-buffer.

Some systems avoid using a dedicated z-buffer by breaking down all intersecting polygons first, allowing intersecting polygons to be simply drawn according to depth order. Compared to systems using a z-buffer, these systems have reduced performance because the CPU must perform the additional polygon partitioning and presorting.

Conventional Texturing and Shading

In addition to hidden-surface removal, when polygons are written into the frame buffer, pixel values must be modified to take account of texture, shading and lighting characteristics.

Textures are stored in memory and relevant pixels from the texture map, called *texels*, are retrieved and used to texture each pixel before it is written into the frame buffer. Depending on the texturing technique, each output pixel requires a different number of texels. The simplest technique, called *point sample texturing*, uses a single texel from the texture map to best approximate a pixel's texture, depending on the orientation of the associated polygon. Point sampling can result in the same texel value being used for several adjacent output pixels, depending on the orientation of the polygon. This technique sometimes causes “blockiness” of the resulting textured polygons.

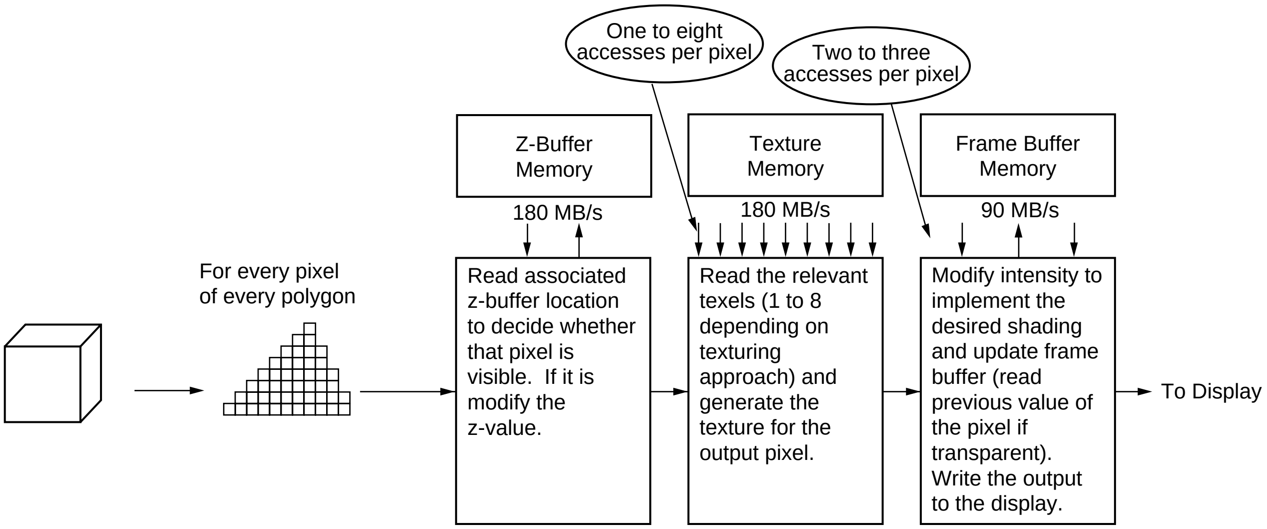
More advanced features interpolate between many texels to more accurately estimate the texture for the output pixel. For example *bilinear texturing* uses four adjacent texels to interpolate the output pixel value, resulting in a smoother-textured polygon as the interpolation smoothes the blockiness associated with point sampling. The disadvantage of bilinear texturing is that it increases the memory bandwidth required to retrieve texture data by up to four times.

Another advanced texturing technique is *MIP mapping*, where low- and high-resolution versions of the same texture are mixed to produce the final value. This technique is particularly useful in removing the aliasing problems that occur when textures are mapped onto acute-angled polygons that disappear into the distance. Aliasing occurs when there are insufficient pixels to faithfully represent sharp variations in the image. Anti-aliasing is important for modeling sky or flat landscapes in which a single resolution is used and image discontinuities occur due to aliasing at certain depths. Varying the mix of low- and high-resolution textures, depending on depth, greatly minimizes this effect.

It is possible to combine point sampling and bilinear techniques within a MIP map approach. *Linear MIP mapping* refers to a case where point sampling is used for both MIP maps when two texels, one from each MIP map, are accessed and combined for every output pixel. *Trilinear texturing* is where bilinear interpolation is used for both MIP maps, resulting in eight texel accesses per output pixel.

Figure 3 shows a basic three-dimensional rendering pipeline, with particular emphasis on memory access requirements. Front-end geometry transformation is omitted for simplicity.

Figure 3. Data Flow in a Simplified Rendering Pipeline


One to eight accesses per pixel
Two to three accesses per pixel
Z-Buffer Memory
Texture Memory
Frame Buffer Memory
180 MB/s
180 MB/s
90 MB/s
For every pixel of every polygon
Read associated z-buffer location to decide whether that pixel is visible. If it is modify the z-value.
Read the relevant texels (1 to 8 depending on texturing approach) and generate the texture for the output pixel.
Modify intensity to implement the desired shading and update frame buffer (read previous value of the pixel if transparent). Write the output to the display.
To Display

In addition to adding textures to polygons, it is often desirable to model surfaces that appear smooth and curved. Processing curved surfaces is very complicated; modifying the pixels' light intensity so that the polygons appear curved and smooth to the observer is a simpler approach that takes into account the position of light sources in the scene relative to the polygon surface. This technique is used to modify the intensity of reflected light at each pixel on the surface, as if the surface normals at each pixel were those of a curved surface passing through vertices of the polygon.

Figure 4. Intensity of Reflected Light at Each Pixel (Smooth Shading)

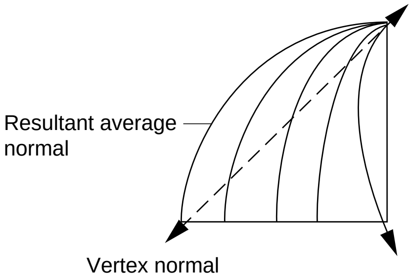


Figure 4 illustrates the basics of smooth shading, where surface normals at vertices are defined and are linearly interpolated (averaged) across each edge and along each display line. The resulting "simulated" curved surface is used in calculating the intensity of reflected light across the polygon. Unlike z-buffering and texturing, the shading operation does not require memory accesses.

Once the output pixel is shaded and textured, it is written into the display frame buffer. If the polygon has any degree of transparency, then the previous value of the same pixel is read from the frame buffer and combined with the new value before being written.

Once all the polygons are processed, the newly generated image is output to the display. This operation is often done in a double-buffered manner to overlap the image display and rendering of the next frame. Double buffering eliminates visual cut lines generated because of the difference between display refresh and rendering frame rate. Figure 3 illustrates the concept of a standard approach to three-dimensional rendering operations including z-buffering, texturing, shading, translucency and frame buffer management.

Limitations of Today's Approaches

Three-dimensional systems using conventional approaches suffer from a number of limitations.

Massive Memory Bandwidth Requirement

There are many accesses into the z-buffer and the texture and frame buffers for each polygon (Figure 3). Assuming a 24-bit z-buffer, 24-bit textures, 24-bit pixels, and a 640 × 480 × 60 Hz refresh rate, these formulas can be used to calculate required bandwidth.

Table 1. Bandwidth Requirements for Conventional Three-Dimensional Systems

Z-Buffer Bandwidth (Bytes/Second)	Texture Bandwidth (Bytes/Second)	Frame Buffer Bandwidth (Bytes/Second)	Total Bandwidth Required
6xPPxPZ	3xPPxPZ (note 3) 6xPPxPZ (note 4) 12xPPxPZ (note 5) 24xPPxPZ (note 6)	3xPPxPZ + 56 MB/sec	12xPPxPZ + 56 MB/sec 15xPPxPZ + 56 MB/sec 21xPPxPZ + 56 MB/sec 33xPPxPZ + 56 MB/sec

Notes:

(1) PZ = polygon size in pixels (average polygon size of 100 pixels)
(2) PP = polygon-per-second performance
(3) Point sampling
(4) Linear MIP mapping
(5) Bilinear interpolation
(6) Trilinear texturing

Table 2 shows the results of these calculations.

Table 2. Examples of Bandwidth Requirements (Average Polygon Size of 100 Pixels)

Case	200k Polygons/Second	500k Polygons/Second	1M Polygons/Second
Point sample	296 MB/sec	656 MB/sec	1256 MB/sec
Linear MIP mapping	356 MB/sec	806 MB/sec	1556 MB/sec
Bilinear interpolation	476 MB/sec	1066 MB/sec	2156 MB/sec
Trilinear texturing	716 MB/sec	1706 MB/sec	3356 MB/sec

It is clear from these examples that the bandwidth requirement for conventional three-dimensional rendering systems is substantial. Furthermore, this requirement increases as the performance level or quality of texturing technique improves.

Memory Consumption

Conventional three-dimensional solutions require memory systems for the z-buffer, texture storage and frame buffer. In addition, many implementations use multi-bank memory systems to optimize data transfer since the bandwidth needed for each one of these memories, particularly texture storage and the z-buffer, is very high and increases with performance. Memory subsystems have necessarily become the most costly part of today's high-performance, three-dimensional rendering systems. In fact, the cost of processing devices is often only a fraction of the cost for required memory components. To achieve a cost-effective three-dimensional graphics solution, therefore, it is essential to eliminate memory requirements wherever possible.

Performance for Cost Sensitive systems Limited by Memory Bandwidth

Given their memory bandwidth and cost, conventional three-dimensional systems cannot deliver both low cost and high performance. Low-cost systems must use cost-effective DRAM memories and combine z-buffer, texture memory and even frame buffer memory wherever possible to share the expensive resources. It is therefore not surprising that systems making such compromises do not deliver much performance. In fact, some suppliers reduce polygon size to a useless 10 to 20 pixels to be able to quote a *high* polygon-per-second figure. This forces the developer to discard details such as textures and polygons to accommodate poor performance. When real games are ported to such solutions, severe performance limits immediately become apparent.

Poor Platform Scalability

Given the nature of the algorithms used in conventional three-dimensional rendering systems, and the hugely different memory bandwidth requirements, none of today's conventional approaches are scalable. It is therefore impossible to use the same solution for a low-end consumer application and a high-end arcade platform. Such scalability is highly desirable because the very best titles are often developed on arcade platforms and then migrated to a consumer PC implementation

Limited Special Effects and Realism

Some of the very powerful and important special effects need a fundamentally different approach in order to achieve low overhead and cost effective implementations. Examples of such capabilities are generation of full shadows where any object can cast shadows on any other object or real time search lights which bring a new dimension to games. Such features cannot be effectively implemented using the conventional polygon based approach. Mechanisms that can better model the real world are needed to address these issues.

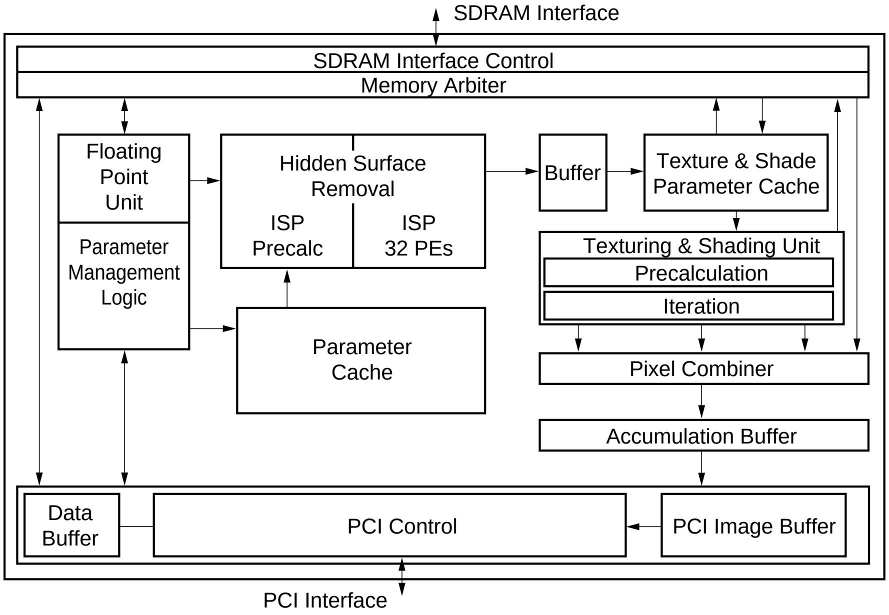
PowerVR Breaks All Conventional Three-Dimensional Barriers

A traditional three-dimensional approach cannot provide both low cost and high performance. PowerVR technology, a joint development between VideoLogic and NEC Corporation, minimizes the amount of memory used *and* the bandwidth of the memory system. As shown in Figure 3, in conventional three-dimensional approaches, two memory subsystems (texture memory and frame buffer memory) hold *real* image data, while the third (z-buffer) temporarily holds depth history during rendering of each frame.

Architecture and Design Objectives

To minimize memory requirements, PowerVR was designed to eliminate the requirement for z-buffer memory. To ensure that maximum performance is maintained, hidden-surface removal is implemented entirely in hardware. This saves z-buffer costs and removes the bottleneck associated with accessing off-chip memory. Furthermore, since hidden-surface removal is performed fully on-chip, calculations are performed at the clock speed of the device and can benefit from continuous improvements to silicon technology.

Figure 5. µPD62011 - A single chip PowerVR implementation



PowerVR also substantially reduces bandwidth requirements of the texture and frame buffers. Texture bandwidth is minimized using *deferred texturing*. Unlike a conventional rendering system where every polygon is textured and the potentially overwritten, the PowerVR approach only textures finally visible pixels (polygons or portions of polygons).

This deferral is possible because the ISP, through innovative hidden-surface removal techniques, outputs pixel groups (tiles) fully resolved in depth and constituting visible pixels only. During the rendering of each frame, every pixel on the screen is only textured once - for the same reason that each output pixel is written to the frame buffer only once. Conversely, in a conventional system, every pixel on every rendered polygon potentially must be written into the frame buffer.

Table 3 uses the same parameters and conditions defined earlier to calculate bandwidth requirements for each of the three main operations in PowerVR.

Table 3. Bandwidth Requirement (Sustained) For PowerVR

z-Buffer Bandwidth (Bytes/Second)	Texture Bandwidth (Bytes/Second)	Frame Buffer Bandwidth (Bytes/Second)	Total Bandwidth Required
0	3xFRxRES (note 3)	3xFRxRES	6xFRxRES + 56M
0	6xFRxRES (note 4)	+	9xFRxRES + 56M
0	12xFRxRES (note 5)	56 MB/sec	15xFRxRES + 56M
0	24xFRxRES (note 6)		27xFRxRES + 56M

Notes:

(1) RES = screen resolution (640 × 480)

(2) FR = rendering frames rate (330)

(3) Point sampling

(4) Linear MIP mapping

(5) Bilinear interpolation

(6) Trilinear texturing

Table 4 shows the results obtained by applying these equations to 200K, 500K and 1M polygons for a 30 Hz rendering frame rate.

Table 4. Example of Bandwidth Requirements (Average Polygon Size of 100 Pixels)

Case	200k Polygons	500k Polygons	1M Polygons
Point sampling	112 MB/sec	112 MB/sec	112 MB/sec
MIPmapping	139 MB/sec	139 MB/sec	139 MB/sec
Bilinear interpolation	194 MB/sec	194 MB/sec	194 MB/sec
Trilinear texturing	305 MB/sec	305 MB/sec	305 MB/sec
Anisotropic filtering	610 MB/sec	610 MB/sec	610 MB/sec

The comparison between the results of Table 2 and Table 4 clearly show two points:

- PowerVR's memory bandwidth requirements are between three to ten times lower than traditional three-dimensional systems
- PowerVR's memory bandwidth requirement for any given texturing approach is independent of polygon-per-second performance figure.

Note that current PowerVR implementations can improve the bandwidth further by using 16-bit textures, and 16-bit frame buffer with the texturing and shading operations all calculated in 24bit and then dithering the output. Also the Z compare is 32-bit which is more accurate than the example conditions without any increase in bandwidth requirement.

Scalable Architecture

Another key feature of the PowerVR technology is its true scalability. PowerVR's display list architecture is inherently scalable. The screen can be segmented and the work shared between PowerVR chips, providing linear performance, allowing system designers to match three-dimensional performance to user requirements.

Hidden surface removal is also key to PowerVR scalability. Hidden surface removal is performed by multiple on-chip processing elements (PEs). The μPD62011 contains 32 parallel PEs. These processing elements operate concurrently and process 32 adjacent pixels. By adding more PEs (in an integrated device or by using multiple devices) it is possible to process more than 32 pixels at a time.

Summary

In summary, traditional 3D rendering approaches rely on large memory systems and performance is limited by memory access speed (bandwidth). This means conventional 3D systems cannot fundamentally deliver both low cost and high performance. PowerVR's reduced memory approach eliminates expensive memory and associated bandwidth. This means PowerVR provides high performance with low cost and ensures scalability as performance is not limited by memory bandwidth and will continue to increase as silicon technology improves.

Phase-out/Discontinued

[MEMO]

[MEMO]

[MEMO]

Phase-out/Discontinued

PowerVR is a trademark of VideoLogic Limited.

No part of this document may be copied or reproduced in any form or by any means without the prior written consent of NEC Corporation. NEC Corporation assumes no responsibility for any errors which may appear in this document.
NEC Corporation does not assume any liability for infringement of patents, copyrights or other intellectual property rights of third parties by or arising from use of a device described herein or any other liability arising from use of such device. No license, either express, implied or otherwise, is granted under any patents, copyrights or other intellectual property rights of NEC Corporation or of others.

M4A 96.10

For further information, please contact:

NEC Corporation
NEC Building
7-1, Shiba 5-chome, Minato-ku
Tokyo 108-8001, Japan
Tel: 03-3454-1111
Fax: 03-3798-6059

[North & South America]

NEC Electronics Inc.
2880 Scott Blvd.
Santa Clara, CA 95050-2554, U.S.A.
Tel: 408-588-6000
800-366-9782
Fax: 408-588-6130
800-729-9288

[Regional Sales Offices]

Central Region
Greenpoint Tower
2800 West Higgins
Road Suite 765
Hoffman Estates,
IL 60195, U.S.A.
Tel: 847-839-6300
Fax: 847-519-9329

Norcal Region
3033 Scott Blvd.
Santa Clara, CA 95054, U.S.A.
Tel: 408-588-5100
Fax: 408-588-5134

Eastern Region
901 N. Lake Destiny Drive
Suite 320
Maitland, FL 32751, U.S.A.
Tel: 407-875-1145
Fax: 407-875-0962

Western Region
One Embassy Centre
9020 S.W. Washington
Square Road
Suite 400
Tigard, OR 97223, U.S.A.
Tel: 503-672-4500
Fax: 503-643-5911

NEC do Brasil S.A.
Eletron Devices Division
Rodovia Presidente Dutra, Km 218
Guarulhos-SP-Brasil
CEP 07210-902
Tel: 011-6465-6810
Fax: 011-6465-6829

[Europe]

NEC Electronics (Germany) GmbH
Kanzlerstr. 2,
40472 Düsseldorf
Germany
Tel: 0211-650302
Fax: 0211-6503490

Munich Office
Arabellastr. 17
81925 München, Germany
Tel: 089-921003-0
Fax: 089-92100315

Stuttgart Office
Industriestr. 3
70507 Stuttgart, Germany
Tel: 0711-99010-0
Fax: 0711-99010-19

Hannover Office
Königstr. 12
30175 Hannover, Germany
Tel: 0511-33402-0
Fax: 0511-33402-34

Benelux Office
Boschdijk 187a
5612 HB Eindhoven,
The Netherlands
Tel: 040-2445845
Fax: 040-2444580

Scandinavia Office
P.O. Box 134
18322 Taeby, Sweden
Tel: 08-6380820
Fax: 08-6380388

NEC Electronics (UK) Limited
Cygnus House, Sunrise Park Way,
Milton Keynes, MK14 6NP, U.K.
Tel: 01908-691-133
Fax: 01908-670-290

NEC Electronics (France) S.A.
9, rue Paul Dautier-BP 187
78142 Velizy-Villacoublay Cédex
France
Tel: 01-30-67-58-00
Fax: 01-30675899

Madrid Office
Juan Esplandiu, 15
28007 Madrid, Spain
Tel: 01-504-2787
Fax: 01-504-2860

NEC Electronics Italiana s.r.l.
Via Fabio Filzi, 25/A,
20124 Milano, Italy
Tel: 02-667541
Fax: 02-66754299

[Asia & Oceania]

NEC Electronics Hong Kong Limited
12/F., Cityplaza 4,
12 Taikoo Wan Road, Hong Kong
Tel: 2886-9318
Fax: 2886-9022/9044

Seoul Branch
10F, ILSONG Bldg., 157-37,
Samsung-Dong, Kangnam-Ku
Seoul, the Republic of Korea
Tel: 02-528-0303
Fax: 02-528-4411

NEC Electronics Taiwan Ltd.
7F, No. 363 Fu Shing North Road
Taipei, Taiwan, R. O. C.
Tel: 02-719-2377
Fax: 02-719-5951

NEC Electronics Singapore Pte. Ltd.
101 Thomson Road #04-02/05
United Square, Singapore 307591
Tel: 65-253-8311
Fax: 65-250-3583

G98. 2

Document No. S11424EJ2V0PF00(2nd edition)
Date Published July 1998 N CP(K)

Printed in Japan