

A 210-mW Graphics LSI Implementing Full 3-D Pipeline With 264 Mtexels/s Texturing for Mobile Multimedia Applications

Ramchan Woo, Sungdae Choi, Ju-Ho Sohn, Seong-Jun Song, and Hoi-Jun Yoo

Abstract—A 121-mm² graphics LSI is designed and implemented for portable two-dimensional (2-D) and three-dimensional (3-D) graphics and MPEG-4 applications. The LSI contains a RISC processor with a multiply-accumulate unit (MAC), a 3-D rendering engine, a programmable power optimizer, and 29-Mb embedded DRAM. The chip is built in a 0.16- μ m pure DRAM technology to reduce the fabrication cost. Texture-mapped 3-D graphics with perspective-correct address calculation and bilinear MIPMAP filtering can be realized while consuming the low power with the help of depth-first clock gating, address alignment logic, and embedded DRAM. Programmable clocking allows the LSI to operate in lower power modes for various applications. The chip consumes less than 210 mW, delivering 66 Mpixels/s and 264 Mtexel/s texture-mapped pixels with real-time special effects such as full-scene antialiasing and motion blur.

Index Terms—Low-power electronics, mobile application, portable, PDA, embedded DRAM, texture mapping, three-dimensional (3-D) graphics rendering.

I. INTRODUCTION

AS THE MOBILE electronics market increases, third-generation (3G) multimedia terminals such as PDAs or smart cell-phones are becoming popular. Their applications are already migrating to the real-time multimedia like MP3 audio, MPEG-4 video [1], [2], and even three-dimensional (3-D) computer graphics [3], [4]. The 3-D applications are especially attractive to games, advertisement, and avatars whose data can be downloaded from the wireless network while occupying only a limited bandwidth. In order to satisfy these market demands, much research on the realization of the 3-D graphics for the handheld devices has recently tried, including the hardware-accelerators designed for mobile platforms [5]–[7] as well as software-only solutions [3], [4]. However, they are still far below the market requirements showing only limited shading operations, without the texture mapping and special rendering effects which are mandatory for the 3-D game applications [3].

Since the realization of real-time 3-D computer graphics requires huge computing power and corresponding memory bandwidth, it has been a critical issue even in PC or console platforms in the past ten years [8]–[10]. It is more challenging on the mobile platform because the power consumption and physical dimension have very stringent limitations. The most

important factor for the handheld devices is low power consumption because of the limited battery lifetime. Based on the allocated budget of system power, the power consumption allowed to the 3-D graphics system is confined to less than 300~400 mW [3]. Also, the 3-D graphics system including the rendering memories must be small in size to be equipped on the limited PCB area of the handheld devices. Therefore, previous graphics processors integrated DRAM on a single die using the embedded memory logic (EML) technology, although it was cost-inefficient due to the process complexity [11].

In this work [12], we designed and implemented a graphics LSI using a pure DRAM technology to reduce the fabrication cost while keeping the high computing power and huge memory bandwidth. Its circuits and architecture are optimized for the real-time application to handheld devices. That is, the full 3-D pipeline is realized with less than 210 mW at the drawing speed of 264 Mtexels/s bilinear MIPMAP textured pixels with special rendering effects. The 3-D graphics images are successfully demonstrated by the fabricated LSI on the PDA evaluation board.

The organization of this paper is as follows. The system architecture will be discussed in Section II, and the design of low-power IP blocks will be covered in Section III. The implementation results of the graphics LSI will be followed in Section IV, and finally, the conclusion of our work will be summarized in Section V.

II. SYSTEM ARCHITECTURE

Fig. 1 illustrates a full 3-D pipeline which covers a geometry engine, a vertex buffer, a rendering engine, and corresponding rendering memories [10]. For real-time 3-D graphics on handheld devices, the geometry engine needs fast calculation of more than 0.5 Mvectors/s and programmability for the transformation and lighting (T&L) [13]. The vertex buffer is necessary for efficient data transfer. The rendering engine requires more than 10-Mpixels/s parallel calculation and more than 1-GB/s huge memory bandwidth for shading, depth comparison, and texturing. Also, a large amount of rendering memory, more than 10 Mb, with high bandwidth reaching to several GB/s, must be prepared to store frame, depth, and various texture images. We developed a 3-D graphics simulator, 3-D-Glamor, to find out an optimum pipeline architecture, memory size, and bandwidth. We gathered the necessary information such as the optimum precision of each datapath, memory bandwidth and utilization, and pipeline efficiency, running various real-time 3-D applications on 3-D-Glamor. Based on its simulation results, we propose the architecture

Manuscript received May 6, 2003; revised September 26, 2003.

The authors are with the Semiconductor System Laboratory, Department of Electrical Engineering and Computer Science, Korea Advanced Institute of Science and Technology (KAIST), Daejeon 305-701, Korea (e-mail: ural@eeinfo.kaist.ac.kr; hjyoo@ee.kaist.ac.kr).

Digital Object Identifier 10.1109/JSSC.2003.821781

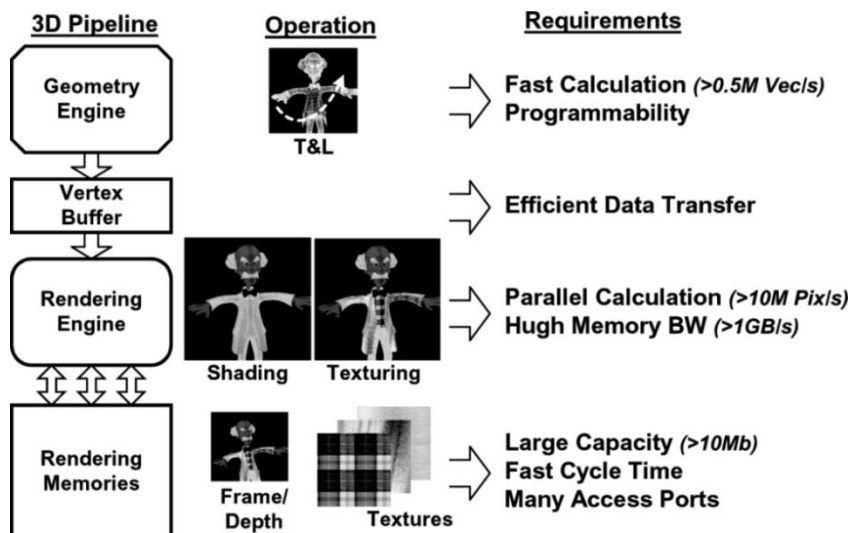


Fig. 1. Integration of full 3-D pipeline.

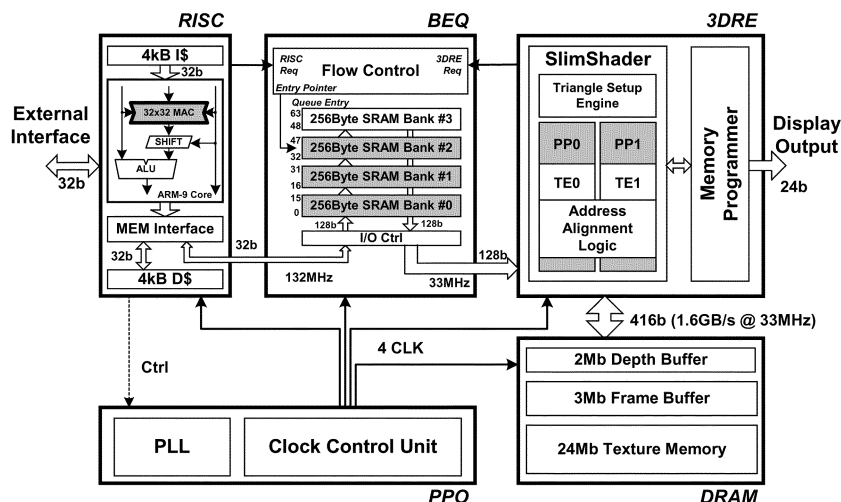


Fig. 2. Block diagram of graphics LSI.

of the graphics LSI as shown in Fig. 2. It consists of a 32-bit RISC processor that is assigned to the geometry engine, a bandwidth equalizer (BEQ) for vertex buffer, a 3-D rendering engine (3-DRE), 29-Mb embedded DRAM, and programmable power optimizer (PPO). Dedicated hardware engines and 1.6-GB/s bandwidth through 416-bit-wide DRAM can lower the operation frequency of 3-DRE even to 33 MHz, while the RISC operates at 132 MHz. Programmable power optimizer manages the power consumption of the chip by controlling four different clock domains—gating the clocks and changing their frequencies during run-time by the software. Each of these IP blocks will be discussed in detail in the next section.

III. LOW-POWER IP BLOCK DESIGN

A. Geometry RISC With Bandwidth Equalizer

The RISC processor with 4 kB I/D caches is compatible with ARM-9 architecture and operates at 132 MHz [14]. It has a single-cycle $32\text{ b} \times 32\text{ b}$ multiply-accumulate unit (MAC) in

its datapath to accelerate the 3-D geometry operations. It can calculate as many as 1.04 Mvertices/s model-view transformations when running a customized fixed-point graphics library, which is a 43% improvement over the conventional ARM-9 processor [13]. Since the conventional 3-D graphics libraries for the PC platforms are optimized to the power-consuming floating-point datapath, they are not suitable for the low-power RISC processor with integer-only datapath. Therefore, we designed a 3-D geometry library with 32-bit fixed-point arithmetic to optimally use MAC-enhanced ARM-9 datapath, maintaining compatibility with OpenGL [15]. The MAC also accelerates the processing of MPEG-4 SP@L1 video stream. It reduces more than 30% of the cycle time when executing the IDCT routines which are basically the same operation as the geometry vector calculation. The memory interface is optimized for real-time multimedia applications so that the RISC can directly supply the 3-D data to the rendering engine through the BEQ, bypassing the data cache.

To compensate the difference of the processing speed and data-width between the RISC and the 3-D rendering engine, the BEQ buffers the vertex data with 1-kB dual-ported SRAM

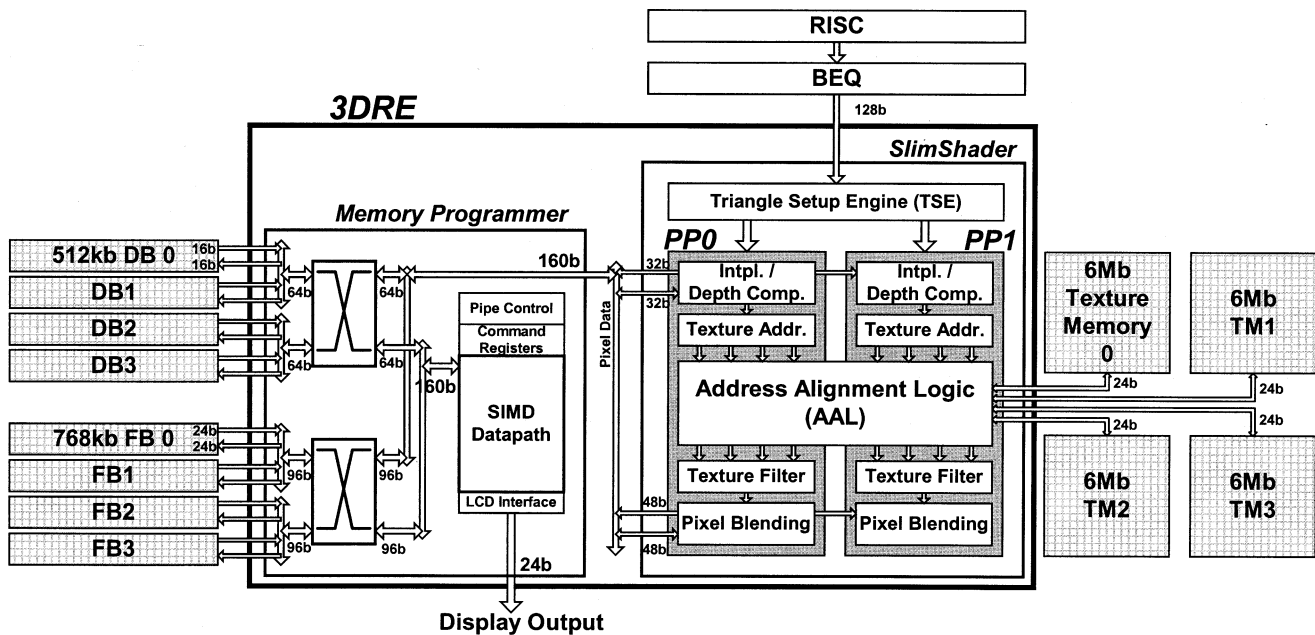


Fig. 3. 3-D rendering engine.

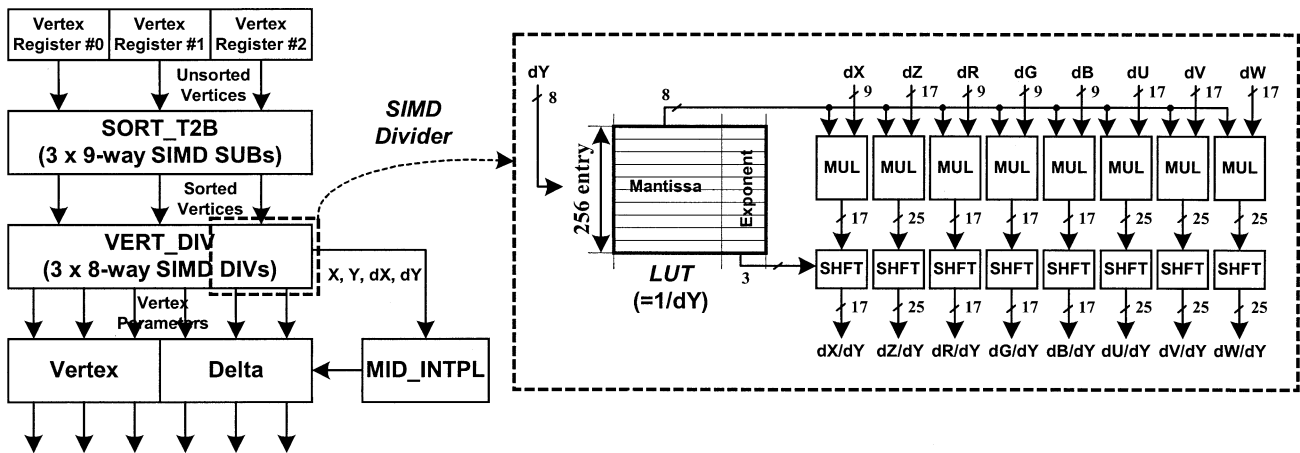


Fig. 4. Triangle setup engine.

(DP-SRAM). The data stored in the vertex buffer are 128 bit-encoded instructions which contain vertex coordinates, texture coordinates, and colors. Revising the previous implementation [7], the current BEQ saves more than 20% power consumption in the SRAM with the help of adaptive bank activation. It partially activates the banks of the DP-SRAM according to the required buffer size, which is decided by the entry pointer. The flow controller (Flow Control) keeps track of the requests from the RISC and the 3-DRE, and activates the only necessary SRAM banks. Since the BEQ is also revised to be configured as 1-kB bidirectional scratch-pad RAM, the RISC can read data from the BEQ for DSP applications in which software-addressable on-chip memory is preferable to store coefficients.

B. 3-D Rendering Engine

Fig. 3 shows the block diagram of the 3-DRE. It consists of a SlimShader, a memory programmer (MP), and a dozen rendering DRAMs. The Slimshader performs the main rendering operations such as texturing, shading, blending, and

depth comparison. The MP enables special rendering effects such as antialiasing, motion blur, and fog to be programmable by the software. The 29-Mb rendering DRAMs contain frame buffers, depth buffers, and texture memories. Twelve independently controlled DRAMs reduce the power consumption since the only necessary memories can be selectively activated. The 3-DRE can accelerate the drawing of points, lines, and rectangles for 2-D graphics as well.

Although triangle setup took more than 7000 cycles when it was calculated by the general-purpose RISC processor, the previous work [5]–[7] did not contain the hard-wired setup engine because of its logic complexity. In this work, however, we simplify the algorithm and implement the triangle setup engine (TSE) which contains three 9-way SIMD SUBS, three 8-way SIMD DIVS, and a midpoint-interpolation unit inside of the 3-DRE to enhance the overall 3-D performance as shown in Fig. 4. SORT_T2B sorts three vertices from top to bottom by subtracting each vertex and checking the sign of the results. Then, VERT_DIV calculates $\Delta(X, Z, R, G, B, U, V, W)/\Delta Y$. At the last stage, MID_INTPL checks the type of triangle by

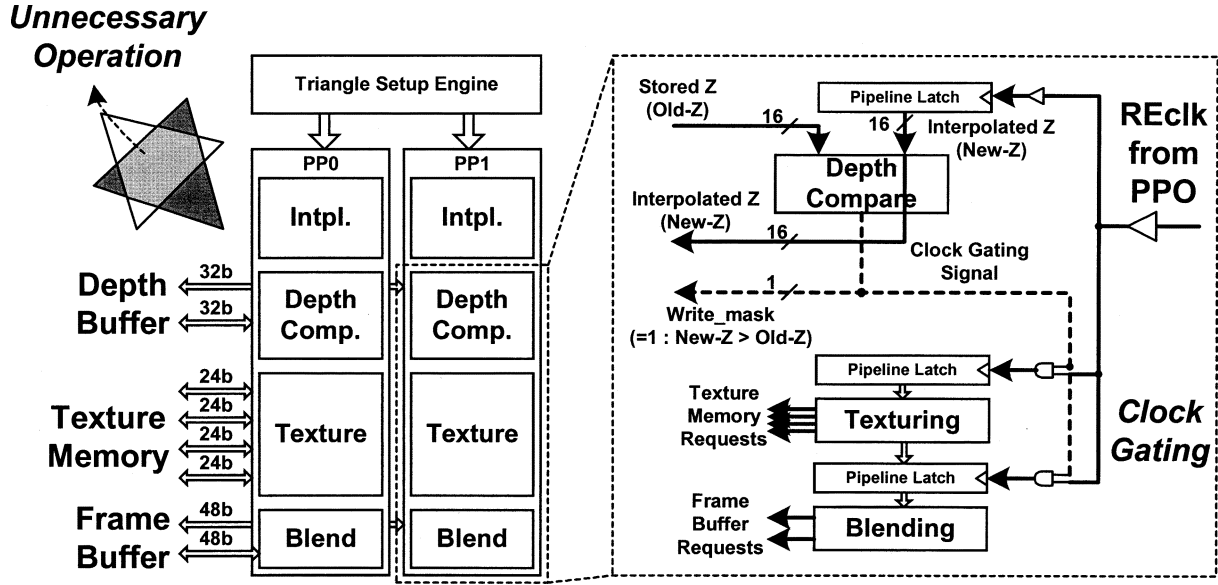


Fig. 5. Depth-first clock gating.

comparing the midpoint of longest edge with the interpolated point.

Once the colors and coordinates are fed into the rendering engine, they are calculated and stored as fixed-point numbers. However, when division operations are necessary, the data are temporally converted to floating-point numbers since the insufficient precision in the fixed-point datapath may result in severe artifacts in the drawing of large polygons. Then the results of division return to fixed-point numbers. For the floating-point division, we simply design the 8-way SIMD divider by using eight integer multipliers, eight shifters, and one precision-controlled look-up table (LUT). All leading zeros are eliminated and only meaningful 8-bit mantissa after the leading zeros and 3-bit corresponding fractional point locations are stored in the LUT. This precision-controlled LUT divider saves power and area by 95% and 85%, respectively, compared with the proprietary IEEE-754 single-precision floating-point divider while delivering the required precision (17 bits for colors or screen coordinates, 25 bits for depth or texture coordinates) for the setup operation.

After the triangle setup operation, pixel data are interpolated [5] and depth-compared within each pixel processor (PP) of SlimShader as shown in Fig. 5. We put the depth-compare unit into the earlier pipeline stage and apply a depth-first clock-gating (DFCG) scheme in order to reduce the power consumption inside the PP. If a new pixel to be drawn is already covered by the nearest pixels from the view point, the new pixel does not need to be processed further. DFCG can prevent the unnecessary shading and texturing by gating off the clock in the remaining datapath according to the results of the depth comparison. It also eliminates the unnecessary requests to the corresponding memories.

C. Low-Power Texturing Unit

Even though the screen resolution of the target PDA is limited, the rendering quality itself cannot be sacrificed. The rendering engine must calculate the pixels correctly within the boundary of the required power budget. Therefore, the 3-DRE contains two texture units, each of which supports

perspective-correct address calculation and bilinear MIPMAP texture filtering. In the calculation of perspective-correct texture addresses, per-pixel division is required. This operation can be described as in the following equations [17]:

$$U = u/w \quad \text{and} \quad V = v/w \quad (1)$$

$$0 \leq (U, V) \leq 1 \quad (2)$$

$$w \geq u, \quad w \geq v \quad (3)$$

where (u, v, w) and (U, V) are homogeneous texture addresses and texture addresses, respectively.

Since each operand $(u, v, \text{ and } w)$ has 16-bit precision in the datapath, 16-bit/16-bit divider is required to calculate the perspective-correct texture addresses $(U \text{ and } V)$. However, by the definition of the texture address as written in (2), the range of w can be limited as in (3). These facts can be used to reduce the power consumption and the area of the address calculation circuit. The w can be represented in a binary form as the composition of leading zeros, 8-bit data, and least significant bits (LSBs). We use only this 8-bit mantissa data to search in the LUT since the leading zeros are meaningless. It reduces the divisor bitwidth from 16 to 8, resulting in more than 95% area reduction in the divider if we can sacrifice the image quality within the 0.78% error boundary, which is quite tolerable to the naked eye, by rounding off the LSBs. Before being fed into the LUT divider, u and v are also reformatted to match w , which is done by left-shifting them by the same number of leading zeros as w and padding zeros after the LSBs.

Eight texel requests are generated at every cycle because two texture units perform the bilinear MIPMAP texture filtering to draw more realistic images [16]. Fetching eight texels directly from eight texture memories (TMs) may consume a large amount of power due to the concurrent data transitions in many capacitive I/Os and the activation power of TMs themselves. Therefore, we adopt address alignment logic (AAL) to reduce the number of memory requests, as illustrated in Fig. 6. Because four texel requests are generated by each pixel processor in the bilinear MIPMAP filtering [16], the total number of requests is

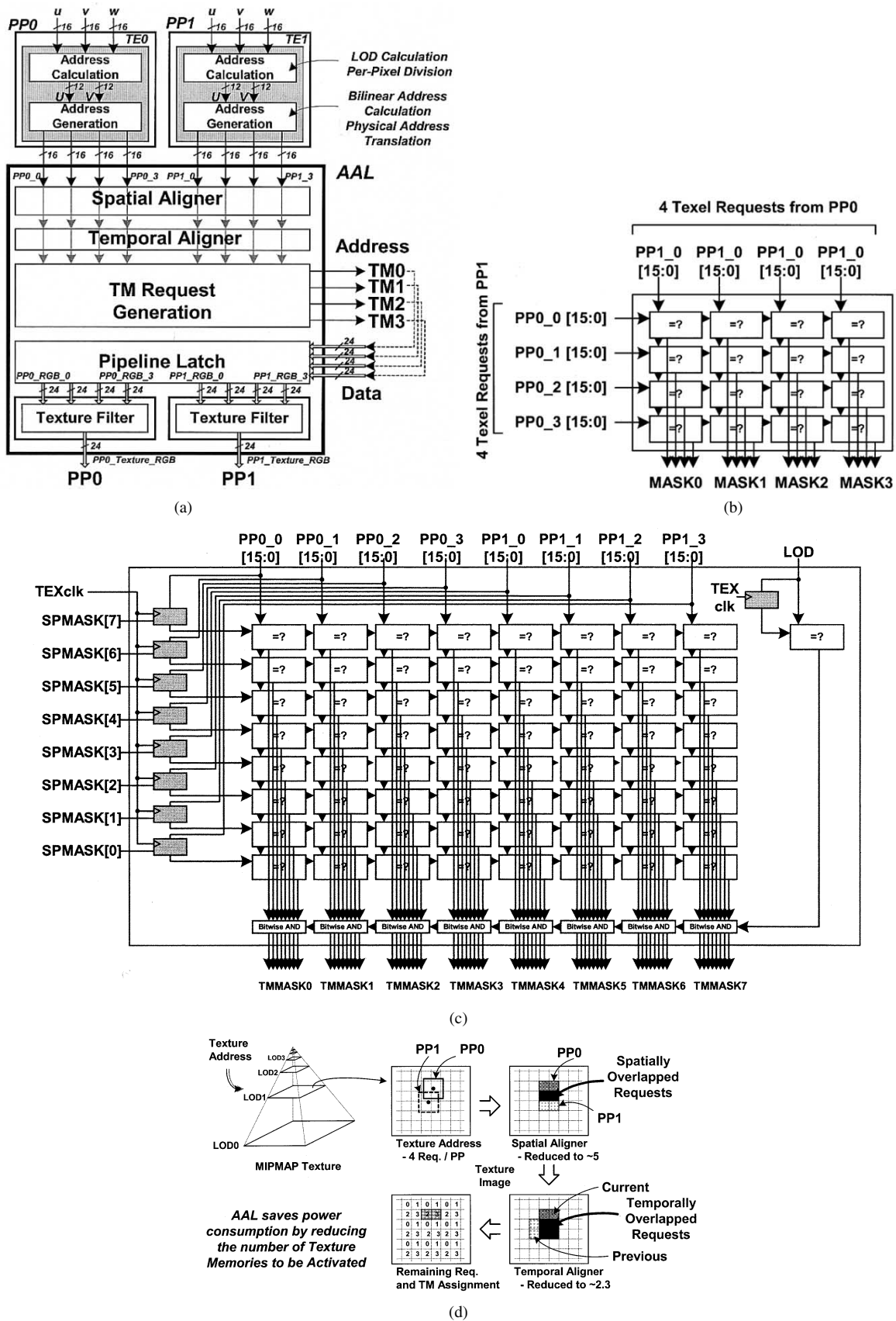


Fig. 6. Address alignment logic. (a) Block diagram. (b) Spatial aligner. (c) Temporal aligner. (d) Operation.

eight. However, there are several requests that are overlapped because their footprints are separated by approximately 1-texel

distance. The spatial aligner finds out and eliminates these overlapped requests reducing the number of requests to five on

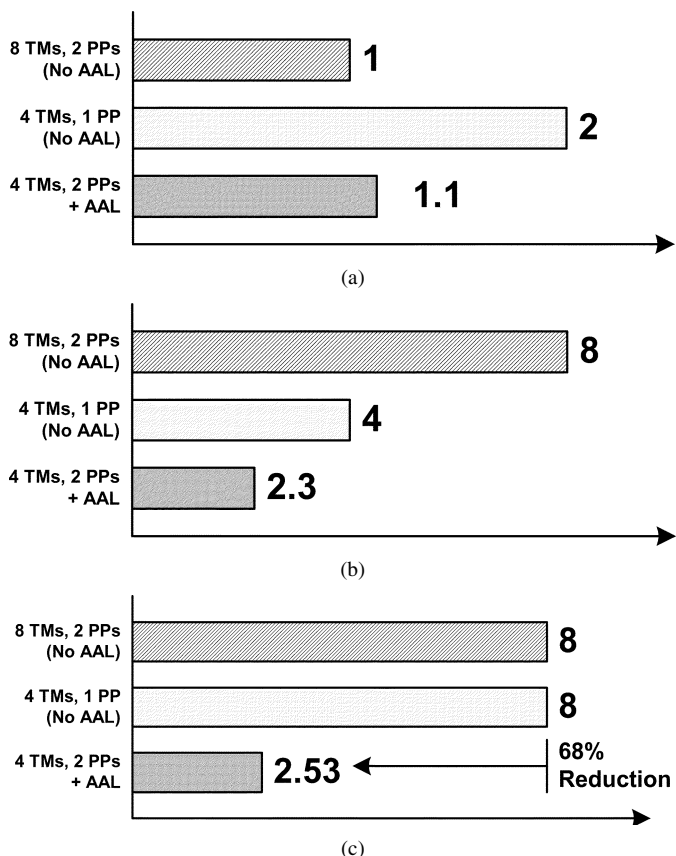


Fig. 7. AAL simulation results. (a) Number of cycles (= time). (b) Number of texture memories activated per cycle (= power). (c) Number of cycles \times number of TM activation (= energy).

an average with 16 comparators as shown in Fig. 6(b). Then, the temporal aligner compares the current texture address with previous ones and leaves only the different addresses. It stores recently used texels working with pipeline latches and comparators as shown in Fig. 6(c). The temporal aligner is basically similar to the 8-entry texture cache [18]. In our architecture, however, texels are simply stored in the pipeline latches instead of power-consuming SRAM. Also, the caching concept is extended to dual pixel processors. After the spatial and temporal overlapping of texels are removed, the average number of remaining requests is reduced to less than 2.3. This means that all of the texels can be fetched from a maximum of four TMs instead of eight although the total number of requests from the pixel processors is eight. A texture image is stored across the texture memories, where adjacent texels are assigned to different texture memory.

This AAL reduces the energy required to draw a scene as summarized in Fig. 7. We gathered the results with 3-D-Glamor, running several benchmarks which are animated on 256×256 screen. Fig. 7(a) shows the number of cycles required to draw two bilinear-filtered pixels, which is proportional to the time required to complete the drawing of a scene. The average number of cycles in the four TMs with AAL is slightly increased to 1.1 due to the memory conflict. Fig. 7(b) shows the power consumption required to activate the texture memories, which is proportional to the number of texture memories to be activated per cycle. With the help of AAL, the number is reduced to 2.3 while doubling the performance compared with four TMs with

TABLE I
CHARACTERISTICS OF EMBEDDED DRAM MACRO

	Frame Buffer	Depth Buffer	Texture Memory
T_{RC}	20ns		
Macro Size	768Kbit	512Kbit	6Mbit
I/O Interface	24bit read 24bit write	16bit read 16bit write	24bit I/O
Commands	Read-Modify-Write Read Write Auto Refresh		Read Write Auto Refresh
Latency	0	0	1

1-PP architecture. Therefore, the energy consumption required to access the texture memory, which is the multiplication of time by power, can be reduced by 68% on average as illustrated in Fig. 7(c). The AAL is followed by a bilinear texture filter which blends four texels into one at every cycle. If point sampling is turned on, each pipeline fetches only one texel instead of four, bypassing the texture filters stage. Finally, the pixel blending stage of SlimShader performs alpha and texture blending operations supporting the OpenGL. The frame buffer stores 24-bit RGB colors without alpha channel. Since the target system is mobile devices, alpha is per-vertex based instead of per-pixel to reduce the memory size.

For real-time special rendering effects, the MP postprocesses the rendered pixels, transferring them to the display controller in parallel with the SlimShader. It contains crossbar switches for front/back buffer sections, and an SIMD-parallel datapath which is controlled by its own 16-bit commands. Since each memory has a separated read/write bus, the total bitwidth of crossbar is 160. The LCD interface reads out the pixels from the front buffer through the SIMD datapath and writes back to the buffer, while the SlimShader performs rendering operations with the back buffer. The postprocessing does not slow down the pixel throughput because the MP processes one pixel per single LCD clock period. The special effects such as full-scene antialiasing, motion blur, and fog can be programmed by the software and downloaded to the command registers. Full-screen antialiasing (FSAA) is performed by 2×1 filtering, and linear fog is calculated with a double depth buffer. The following equations are examples of post-filters which can be evaluated by SIMD datapath.

$$\text{FSAA: } \text{OUT}[x][y] = (a * \text{FB}[x][y] + b * \text{FB}[x+1][y]) / c$$

(for example, $a = 3, b = 1, c = 4$)

$$\text{Fog: } \text{OUT}[x][y] = a * (\text{FB}[x][y] - \text{color}) + \text{color}$$

$$(a = (\text{ZB}[x][y] + \text{bias} / \text{SCREEN_DEPTH}),$$

$$0 < a < 1 \text{ saturated})$$

D. 29-Mb Embedded DRAMs

To save the power consumption of the embedded DRAMs as well as to optimally utilize their bandwidth, we design three different DRAM types. As described in Table I, the characteristics of each memory are optimized according to its operation requirements.

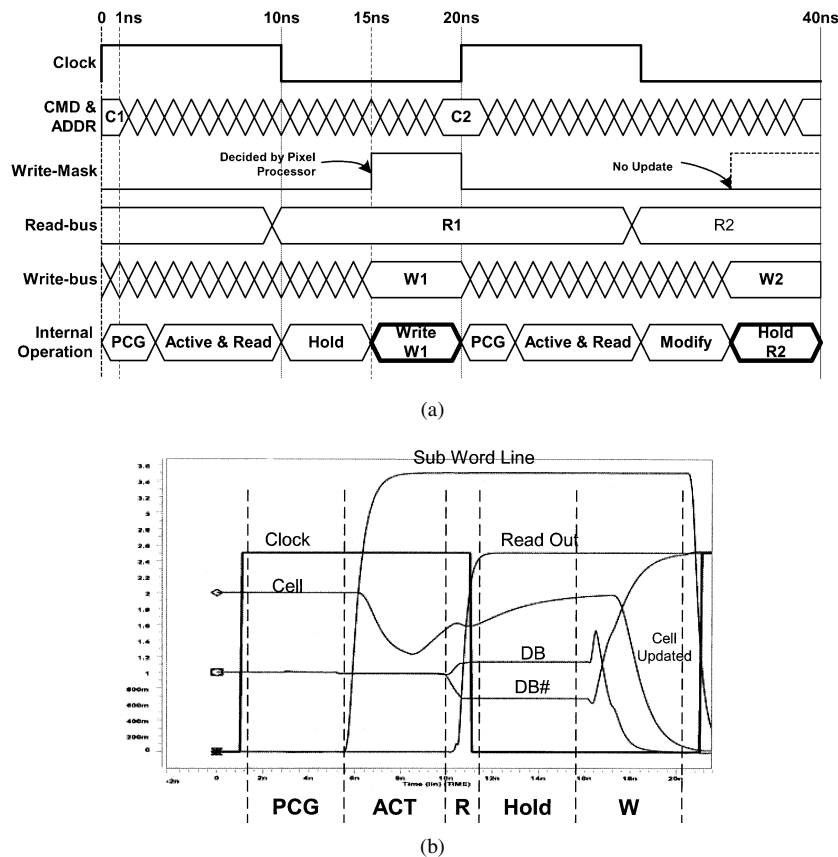


Fig. 8. Frame buffer access timing. (a) Timing diagram. (b) Simulation waveform.

In order to provide the pixels for depth comparison and alpha blending, the frame and depth buffers support a read-modify-write data transaction in a single cycle with a separated read and write bus. It drastically simplifies the memory interface of the rendering engine and the pipeline, because the data required to process two pixels are read from the frame and depth buffers, calculated in the pixel processor, and written back to the buffers within a single clock period without any latency. Therefore, caching and prefetching, which may cause power and area overhead, are not necessary in this architecture. The operation timing and the simulation waveform of frame buffer are shown in Fig. 8. The Write-Mask signal, which is generated by the pixel processor, decides the activation of the write operation. Nonmultiplexed addressing enables the DRAM to partially activate the necessary wordline block to save the power consumption inside the memory [5]–[7]. To draw pixels on the 256×256 screen, which covers the resolution of most of the current cell phones, four frame macros and four depth macros are used in the chip. Also, four texture memory macros, or 24 Mb, store MIPMAP texture images for 3-D game applications.

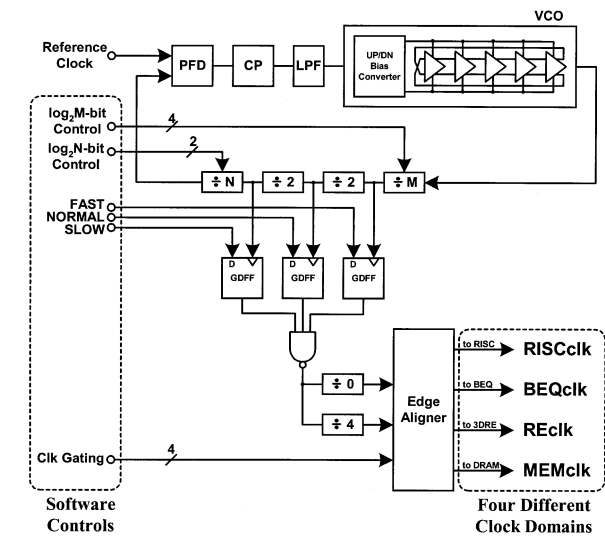
E. Programmable Power Optimizer

The PPO manages the power consumption of the chip. Each clock can be selectively gated and its frequency is scalable by the software to adjust the frame rate during run-time. RISCclk and BEQclk run at the full speed of the RISC core, and REclk and MEMclk operate at the quarter frequency—132/33 MHz (RISCclk/REclk) for FAST mode, 66/16.5 MHz for NORMAL, and 33/8.25 MHz for SLOW. The circuit diagram of the PPO is

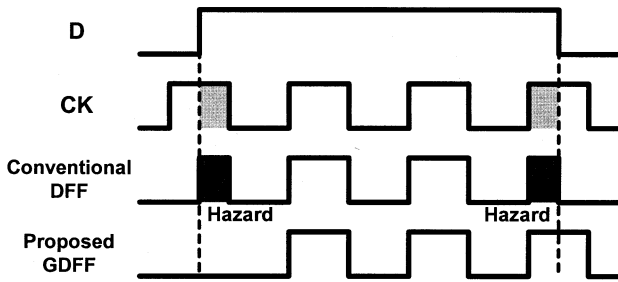
shown in Fig. 9(a). It provides zero-latency frequency scaling to allow abrupt switching of operating frequencies during the execution of software. When switching the speed modes, a glitch may occur due to the inherent skew in the gating logic as shown in Fig. 9(b). In addition, a simple blending of the scaled frequencies from feedback frequency dividers may generate a surge spike on the clock signals. In order to avoid such a glitch or surge in the clock signal, the gated D-flip-flop (GDFF) is proposed as illustrated in Fig. 9(c). Therefore, as shown in the measured waveforms in Fig. 9(d), the transition from slow mode to fast mode can be completed quickly without any hazard. The PPO containing the phase-locked loop (PLL) consumes less than 3-mW power.

IV. IMPLEMENTATION AND MEASUREMENT RESULTS

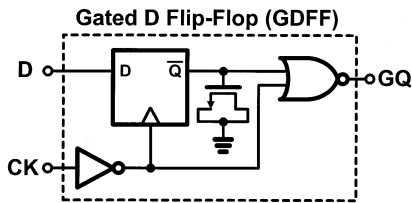
To implement the portable 3-D graphics LSI, previous chips integrated DRAM using EML technology [5]–[7]. However the fabrication process costs too much because the logic must be designed with separate transistors from the DRAM with more mask layers. Therefore, low-cost mobile platforms have not widely used EML technology yet. In this work, we implement the LSI with the pure DRAM process instead of the EML to reduce the fabrication cost. The logic components, SRAM, and analog blocks are drawn with the design rule of peripheral transistors of the DRAM. All logics are synthesized with DRAM-optimized standard-cell library. They meet the performance requirements of mobile applications although they show relatively long gate delay and large routing area compared with the pure logic process [12]. This DRAM-based



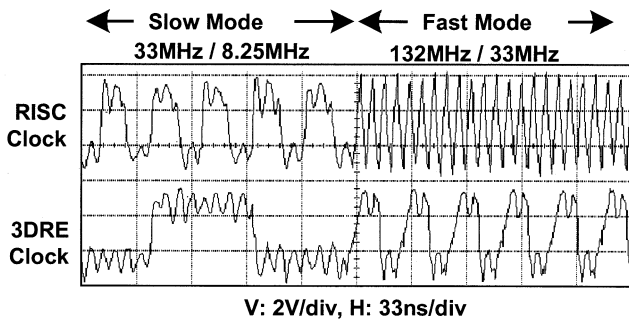
(a)



(b)



(c)



(d)

Fig. 9. PPO circuits and measurement results. (a) PPO circuits. (b) Hazards when clock changes. (c) Circuit diagram of GDFF. (d) Mode transition from slow to fast.

SoC implementation enables us to use large on-chip memory with inherently little leakage current, which is important for mobile multimedia applications. Since this LSI is fabricated with the same as 256-Mb DRAM process, the subthreshold leakage is negligible.

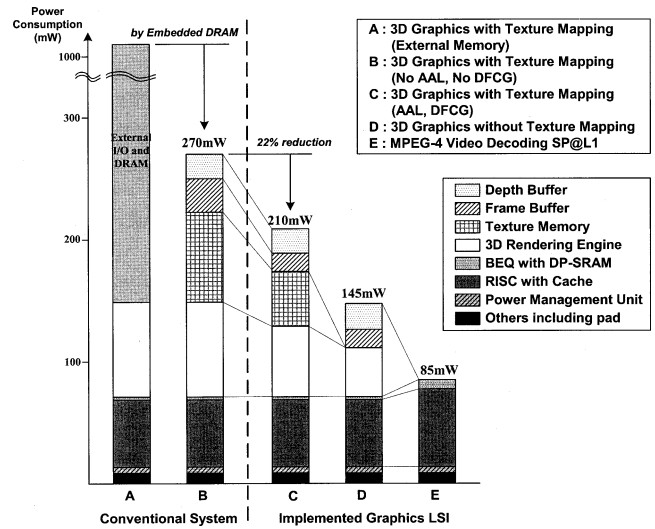


Fig. 10. Power consumption of graphics LSI.

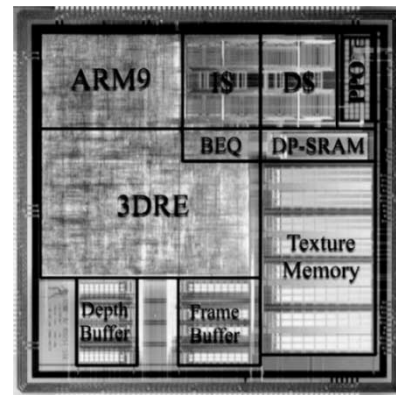


Fig. 11. Die photograph.

Fig. 10 shows the composition of the power consumption for various applications. The implemented graphics LSI consumes 210 mW in continuous calculation of bilinear texture-mapped and antialiased 3-D graphics applications at FAST mode (33-MHz REclk and 132-MHz RISCclk). The embedded DRAM drastically reduces the power consumption since the external I/Os for 3-D rendering are eliminated, and an additional 22% reduction is obtained by AAL and DFCG. For point-sampled texturing, the power reduces to 185 mW. Nontextured (but Gouraud-shaded) 3-D applications and MPEG-4 video decoding consume 145 mW and 85 mW, respectively. Textured 3-D rendering consumes 110 mW at NORMAL (16.5-MHz REclk and 66-MHz RISCclk), and 65 mW at SLOW mode (8.25-MHz REclk and 33-MHz RISCclk), respectively. The power consumption of MP is about 5 mW, which is low because it is synchronized with 10-MHz LCD clock.

Fig. 10 compares the performance of the proposed SlimShader with the previous architectures [5]–[7]. Just pixel or texel fill rate are insufficient to indicate the rendering performance of mobile applications because the power consumption must be considered as well. Based on the performance indices of portable 3-D graphics [6], the pixel rate of this LSI is about 0.8-MPXPS/mW, which is 1.6 times greater than that of the previous work. The texel rate is about 1.88-Mtexels/s per

TABLE II
CHARACTERISTICS OF THE GRAPHICS LSI

Process Technology	0.16 μ m CMOS DRAM with 1-W 3-AI
Power Supply	2.0V (DRAM Core), 2.5V (Logic), 3.3V (I/O)
Operating Frequency (RISC, BEQ/3DRE, DRAM)	FAST : 132MHz/33MHz NORMAL : 66MHz/16.5MHz SLOW : 33MHz/8.25MHz
Power Consumption	< 210mW
Transistor Counts	1M Logic 29Mbit DRAM 72kbit SRAM (9KByte)
Die Size	11mm x 11mm
Package	240pin QFP
Target Applications	Realtime 2D/3D Graphics Pipeline MPEG-4 SP@L1 Decoding MP3 Audio Decoding
Embedded DRAM	5Mb Double Depth / Frame Buffer (256 x 256 Resolution, 24bit Color, 16bit Depth) 24Mb Texture Memory
3D Geometry Performance with Fixed-Point Graphics Library	1.04Mvertices/s : Model-View Transformation 300kvertices/s : Model-View Transformation + Perspective Projection + 6-Side Clipping 70kvertices/s : Model-View Transformation + Perspective Projection + 6-Side Clipping + Lighting (Single directional light source from infinite viewer, one-sided, ambient + diffuse + specular highlighting)
3D Rendering Performance	66Mpixels/s, 264Mtexels/s Hardware Triangle Setup Engine Perspective-Correct Bilinear MIPMAP Texturing Gouraud Shading, Alpha Blending, Texture Blending Antialiasing, Motion Blur, Fog, Special Effects

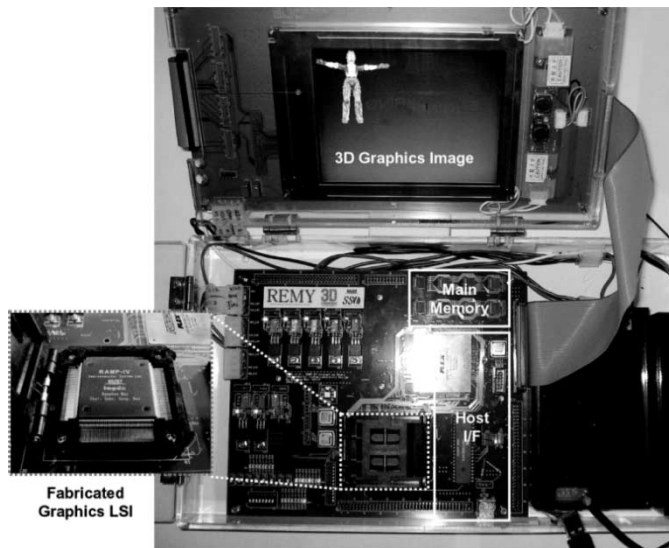


Fig. 12. Demonstration on PDA evaluation board.

milliwatt (MTXPS) which is, to the best of our knowledge, the highest ever published for portable devices.

The graphics LSI is implemented using a typical 0.16- μ m DRAM process with 1-W 3-AI metal layers and its die area takes 121 mm². The chip contains 1 M logic transistors, 29-Mb DRAM, 72-kB SRAM, and a PLL. Fig. 11 shows the die photograph and Table II summarizes its features. It can draw 24-bit texture-mapped pixels at the drawing speed of 66 Mpixels/s and 264 Mtexels/s. The realistic 3-D graphics images with texture mapping are successfully demonstrated by the fabricated chip on the PDA system board, and it is shown in Fig. 12.

V. CONCLUSION

A low-power graphics LSI is designed and implemented for mobile multimedia applications. The LSI contains a 32-bit RISC processor with enhanced MAC, a 3-D rendering engine, a programmable power optimizer, and 29-Mb embedded DRAMs. Full 3-D graphics pipeline featuring 66 Mpixels/s and 264 Mtexels/s texture-mapped 3-D graphics as well as 2-D MPEG-4 video decoding consumes less than 210 mW and 121 mm² chip area. The chip is implemented with 0.16- μ m pure DRAM process to reduce the fabrication cost. The 3-D graphics images are successfully demonstrated using the fabricated chip on the PDA evaluation board.

ACKNOWLEDGMENT

The authors would like to thank Y.-D. Bae, C.-W. Yoon, B.-G. Nam, J.-H. Woo, S.-E. Kim, and I. Park of KAIST, and S. Shin, K.-D. Yoo, and J.-Y. Chung of Hynix Semiconductor for their contributions, and the Memory R&D division of Hynix Semiconductor for the chip fabrication.

REFERENCES

- [1] T. Hashimoto *et al.*, "A 27-MHz/54-MHz 11-mW MPEG-4 video decoder LSI for mobile applications," *IEEE J. Solid-State Circuits*, vol. 37, pp. 1574–1581, Nov. 2002.
- [2] T. Nishikawa *et al.*, "A 60 MHz 230 mW MPEG-4 video-phone LSI with 16 Mb embedded DRAM," in *IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers*, Feb. 2000, pp. 230–231.
- [3] Khronos Group, "Bringing 3-D gaming to cell phones," presented at the Game Developers Conf., San Jose, CA, 2003.
- [4] G. K. Kolli, "3-D Graphics optimizations for ARM architecture," presented at the Game Developers Conf., San Jose, CA, 2002.
- [5] Y.-H. Park *et al.*, "A 7.1-GB/s low-power rendering engine in 2-D array-embedded memory logic CMOS for portable multimedia system," *IEEE J. Solid-State Circuits*, vol. 36, pp. 944–955, June 2001.
- [6] R. Woo *et al.*, "A 120 mW 3-D rendering engine with 6 Mb embedded DRAM and 3.2 GB/s runtime reconfigurable bus for PDA-chip," *IEEE J. Solid-State Circuits*, vol. 37, pp. 1352–1355, Oct. 2002.
- [7] C.-W. Yoon *et al.*, "A 80/20 MHz 160 mW multimedia processor integrated with embedded DRAM, MPEG-4, and 3-D rendering engine for mobile applications," *IEEE J. Solid-State Circuits*, vol. 36, pp. 1758–1767, Nov. 2001.
- [8] S.-J. Park *et al.*, "A reconfigurable multilevel parallel texture cache memory with 75-GB/s parallel cache replacement bandwidth," *IEEE J. Solid-State Circuits*, pp. 612–623, May 2002.
- [9] A. K. Khan *et al.*, "A 150 MHz graphics rendering processor with 256 Mb embedded DRAM," in *IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers*, Feb. 2001, pp. 150–151.
- [10] J. S. Montry *et al.*, "InfiniteReality: A real-time graphics system," in *Proc. SIGGRAPH*, 1997, pp. 293–302.
- [11] D. D. Buss, "Technology in the internet age," in *IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers*, Feb. 2002, pp. 18–21.
- [12] R. Woo *et al.*, "A 210 mW graphics LSI implementing full 3-D pipeline with 264 Mtexels/s texturing for mobile multimedia applications," in *IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers*, Feb. 2003, pp. 44–45.
- [13] J.-H. Sohn *et al.*, "Optimization of portable system architecture for real-time 3-D graphics," in *Proc. IEEE Int. Symp. Circuits and Systems*, 2002, pp. 1769–1772.
- [14] Y.-D. Bae *et al.*, "A single-chip programmable platform based on a multithreaded processor and configurable logic clusters," in *IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers*, Feb. 2002, pp. 336–337.
- [15] OpenGL (2003). [Online]. Available: <http://www.opengl.org>
- [16] L. Williams, "Pyramidal parametrics," in *Proc. SIGGRAPH*, 1983, pp. 1–11.
- [17] P. S. Heckbert, "Survey of texture mapping," *IEEE Comput. Graph. Appl.*, vol. 6, pp. 56–67, Nov. 1986.
- [18] Z. S. Hakura and A. Gupta, "The design and analysis of a cache architecture for texture mapping," in *Proc. 24th Int. Symp. Computer Architecture*, 1997, pp. 108–120.



Ramchan Woo was born on January 1, 1978, in Korea. He received the B.S. (*summa cum laude*) and M.S. degrees in electrical engineering from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, in 1999 and 2001, respectively. He is currently working toward the Ph.D. degree in electrical engineering at KAIST.

In 1999, he joined the Semiconductor System Laboratory (SSL) at KAIST as a Research Assistant. His research interests include low-power, high-performance circuits, and portable multimedia system design with specific interest in mobile 3-D computer graphics architecture and its implementation with merged-DRAM technology. Also, he is now working for the mobile graphics libraries.



Sungdae Choi was born on March 17, 1978, in Korea. He received the B.S. and M.S. degrees in electrical engineering and computer science in 2001 and 2003, respectively, from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, where he is currently working toward the Ph.D. degree.

In 2001, he joined the Semiconductor System Laboratory (SSL) at KAIST as a Research Assistant. His research activities are related to application-specific embedded memory architecture and content-address-

able memories.



Ju-Ho Sohn was born on July 7, 1979, in Korea. He received the B.S. (*summa cum laude*) and M.S. degrees in electrical engineering in 2001 and 2003, respectively, from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, where he is currently working toward the Ph.D. degree in electrical engineering.

His research activities are related to real-time 3-D graphics for portable systems and its implementation, especially high-performance portable multimedia processor design for 3-D vertex operations.



Seong-Jun Song (S'01) was born in Seoul, Korea, in 1979. He received the B.S. degree in electrical engineering and computer science in 2001 from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea, where he is currently working toward the M.S. degree.

Since 2001, he has been a Research Assistant at KAIST. His research interests include high-speed optical interface integrated circuits using submicron CMOS technology, phase-locked loops and clock and data recovery circuits for high-speed data communications, and radio-frequency CMOS integrated circuits for wireless communication applications.



Hoi-Jun Yoo graduated from the Electronic Department of Seoul National University, Seoul, Korea, in 1983 and received the M.S. and Ph.D. degrees in electrical engineering from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, in 1985 and 1988, respectively. His Ph.D. work concerned the fabrication process for GaAs vertical optoelectronic integrated circuits.

From 1988 to 1990, he was with Bell Communications Research, Red Bank, NJ, where he invented the two-dimensional phase-locked VCSEL array, the front-surface-emitting laser, and the high-speed lateral HBT. In 1991, he became Manager of a DRAM design group at Hyundai Electronics and designed a family of fast 1 M DRAMs and synchronous DRAMs, including 256 M SDRAM. From 1995 to 1997, he was a faculty member with Kangwon National University. In 1998, he joined the faculty of the Department of Electrical Engineering at KAIST, and currently leads a project team on RAM Processors (RAMP). In 2001, he founded System Integration and IP Authoring Research Center (SIPAC), a national research center funded by the Korean government to promote worldwide IP authoring and its SoC application. Currently, he is the Project Manager for SoC in the Korea Ministry of Information and Communication. His current interests are SoC design, IP authoring, high-speed and low-power memory circuits and architectures, design of embedded memory logic, optoelectronic integrated circuits, and novel devices and circuits. He is the author of the books *DRAM Design* (Seoul, Korea: Hangleung, 1996; in Korean) and *High Performance DRAM* (Seoul, Korea: Sigma, 1999; in Korean).

Dr. Yoo received the Electronic Industrial Association of Korea Award for his contribution to DRAM technology in 1994 and the Korea Semiconductor Industry Association Award in 2002.