

Low-Power 3D Graphics Processors for Mobile Terminals

Ju-Ho Sohn, Yong-Ha Park, Chi-Weon Yoon, Ramchan Woo, Se-Jeong Park, and Hoi-Jun Yoo, KAIST

ABSTRACT

A full 3D graphics pipeline is investigated, and optimizations of graphics architecture are assessed for satisfying the performance requirements and overcoming the limited system resources found in mobile terminals. Two mobile 3D graphics processor architectures, RAMP and DigiAcc, are proposed based on the analysis, and a prototype development platform (REMY) is implemented. REMY includes a software graphics library and simulation environment developed for more flexible realization of mobile 3D graphics. The experimental results demonstrate the feasibility of mobile 3D graphics with 3.6 Mpolygons/s at 155 mW power consumption for full 3D operation.

INTRODUCTION

The popularity of mobile communications such as smart cell phones and wireless PDAs is increasing with the rapid expansion of the mobile electronics market and its migration from text-based applications to various multimedia applications. Real-time 3D graphics is becoming one of the most attractive applications in mobile terminals due to its benefits for gaming, advertising, and avatars whose data can be downloaded over the wireless network. 3D graphics is also advantageous for bandwidth-constrained wireless applications, as complex 3D scenes can be represented by lists of vertices, texture images, and corresponding camera movements, yielding high-data-compression ratios.

A 3D graphics pipeline is composed of geometry operations for calculating the attributes of vertices of triangles, and rendering operations for filling colors inside the triangles. The geometry stage processes polygon data from input models using operations such as transformation, lighting, and perspective projection. Lighting effects are calculated by blending ambient, specular, diffuse, and emission components originated by each light source. The resulting geometry stage is computation intensive, but the bottleneck can be relieved using fast, parallel datapaths such as multicore vector processors with a 3D graphics-optimized instruction-set architecture. Software simulation indicates that more than 40 GOPS is required when calculating a full

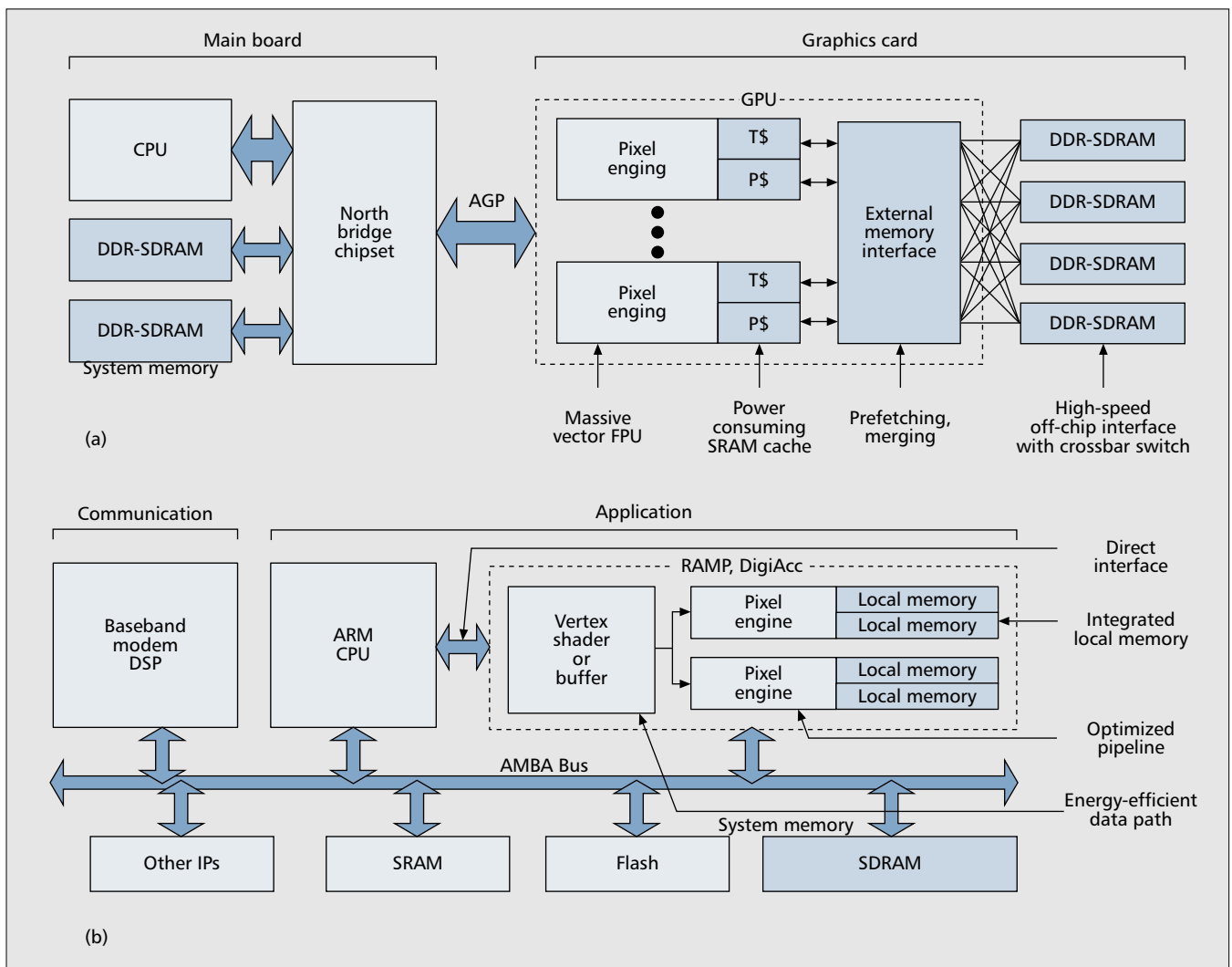
suite of geometry operations at a speed of 1 Mvertices/s in conventional embedded RISC processors [1].

The rendering stage takes the output of the geometry stage and draws pixels to the screen buffer. First, 3D geometry data are transformed into sets of 2D triangles using interpolation to calculate edge coordinates for each triangle. Each pixel is then rendered by shading and texture mapping. The rendering stage also performs alpha blending for translucent objects, and depth comparison for hidden surface removal. Rendering-stage operations are memory-intensive due to frequent accessing of the frame buffer, depth buffer, and texture memory. An effective graphics system memory bandwidth of 1 Gbyte/s is required to show realistic 3D images with a pixel fill rate of 1 Mpixels/s on today's mobile terminals with QVGA screen size.

Mobile 3D graphics are providing more functionality in both hardware and software while achieving low power consumption. Recently, researchers have tried to increase graphics capabilities in mobile applications using techniques such as tile-based rendering and compression to minimize the required memory bandwidth. They have used standard bus interfaces for integrating graphics accelerators in mobile platform system on chip (SoC) and chipset. Various low-power techniques such as clock gating and power-down mode have also been developed. More advanced mobile multimedia solutions are introducing programmability and additional hardware blocks such as MPEG4 encoding/decoding and camera engines in a single chip.

Standard software graphics application programming interfaces (APIs) for embedded systems have also been released. One example is OpenGL-ES, which is subset of desktop OpenGL. OpenGL-ES adopts optimizations such as fixed-point operations and redundancy eliminations for mobile devices with low processing power, while enabling fully programmable 3D graphics such as vertex and pixel shading.

This article describes design issues in mobile 3D graphics. The PC graphics hardware architecture with its shortcomings in mobile environments is described, and architectural optimizations are discussed. Two graphics processors are presented: RAM Processor (RAMP), which utilizes embedded DRAM technology for



■ **Figure 1.** a) Example of PC graphics architecture; b) example of cell phone architecture.

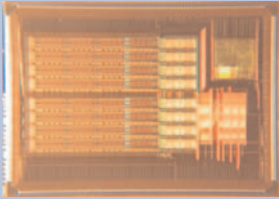

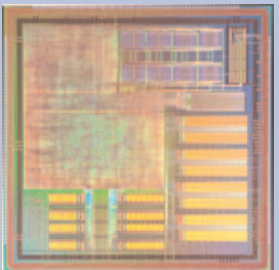
efficient memory organizations, and Digital Accessory (DigiAcc), which utilizes enhanced coprocessor architecture for advanced programmable shading. The REMY prototype development platform is also described. The article concludes with a performance comparison of various graphics architectures.

PC VS. CELL PHONES

The PC graphics architecture, which has its roots in traditional workstation graphics systems, has been applied to various consumer electronics and battery-operated laptop computers for many years. Hence, PC graphics can be used to consider design issues for mobile terminals. Figure 1a shows a typical example of today's PC graphics architecture, in which the graphics-processing unit (GPU) has been evolved to attain huge memory bandwidth and high parallelism [2]. The GPU contains large vector floating-point units (FPUs) with special instructions for graphics operations. The main CPU invokes the GPU using the system bus interface. However, the available system-memory bandwidth is not sufficient to support both the CPU and the GPU. In the GPU architecture, several pixel engines work in parallel to boost performance, fetching data

from dedicated T\$ (texture cache) and P\$ (pixel cache) memories. The external memory interface (EMI) merges transactions from cache memories and transfers them to off-chip DDR-SDRAMs dedicated to graphics processing. The memories are connected to the EMI through a high-speed crossbar switch. Burst-mode operations are used to fully utilize the available memory bandwidth. Although each cache element and FPU can be power-efficient, the massive structure and high-speed crossbar of the GPU cannot be applied directly to mobile terminals that lack sufficient computing power and memory capacity. In addition, modern baseband chips and mobile platforms such as Qualcomm's MSM chip or TI's OMAP employ power-efficient advanced RISC machine (ARM) processors of integer datapath, and are implemented as SoCs (optimized for battery-operated mobile devices).

The RAMP and DigiAcc architectures are proposed as responses to these concerns. They utilize a simple ARM coprocessor interface or dedicated buffer connected to an energy-efficient fixed-point graphics accelerator with specific local memory. Figure 1b shows an example cell-phone architecture using the proposed approaches.

Architecture	Process technology	Power consumption	Die area	Embedded DRAM	3D performance	Design methodology
RAMP-I	 0.35 μm merged DRAM	560 mW	56 mm^2	512 kb	Gouraud — shading, alpha — blending, depth — comparison	Full custom
RAMP-II	 0.18 μm merged DRAM	160 mW	84 mm^2	6 Mb		
RAMP-IV	 0.16 μm pure DRAM	210 mW	121 mm^2	29 Mb	+Bilinear MIPMAP texturing, special rendering effects, antialiasing	RTL synthesized

■ Table 1. Summary of RAMP chips.

RAMP GRAPHICS PROCESSOR

DESIGN PHILOSOPHY

Rendering operations such as rasterization and texture mapping dominate the 3D graphics pipeline, and require high-memory bandwidth [1]. Solving the bandwidth bottleneck with traditional approaches such as high-speed crossbars and off-chip DDR-SDRAMs can result in increased power consumption. However, the limited screen resolutions in mobile terminals (e.g., QVGA) imply that a reasonable amount of integrated memory, from tens of kBytes to hundreds of kBytes, is sufficient for graphics memories when used as high hit-rate caches or frame buffers. In addition, embedding all of the required memory with the logic on a single die yields more efficient architectures and implementations in terms of performance and power consumption.

The RAMP design methodology is based on the philosophy that memory is no longer a *passive* device, nor a *subsystem*. The RAMP (RAM Processor) architecture utilizes embedded DRAM for 3D rendering in a very efficient manner that avoids connecting the memory with a large number of wires and corresponding crossbar switch. Instead, the optimal memory configuration is determined by analyzing the bandwidth requirements and access patterns of the application. The characteristics of 3D rendering operations are exploited to distribute the memory accesses in time, hence reducing the power consumption by activating one or some of the memories locally. The design of the embedded DRAM is specified according to its location on the die and expected access patterns. Hence, the latency, throughput, number of busses, and com-

mands of each DRAM block are treated as application-specific variables. The logic pipelines are tuned to the modified timing and functions of the DRAMs. Various low-power techniques such as clock gating are used extensively inside the memory and logic.

RAMP ARCHITECTURES

Three RAMP chips were evaluated in order to demonstrate the RAMP architectures and methodology. Table 1 summarizes the physical characteristics and supported 3D features of the different chips.

RAMP-I — RAMP-I can draw Gouraud shaded, alpha-blended and depth-compared pixels at a 40Mpixels/s drawing speed on 256×256 screen resolution [3]. The RAMP-I architecture is described in Fig. 2. The proposed 2D HOT (hierarchical octet tree) array structure consists of a preprocessing RISC core, a bus-matching FIFO, eight edge processors (EPs), 64 pixel processors (PPs), 64 DRAMs, and 64 SRAM serial access memories (SAMs). It shows a two-level octet tree structure that is made of a single master processor and eight slave processors at each level. A polygon-based rendering command, supplied with 32-bit bandwidth from the RISC core, is broadcasted over the EPs, and they calculate the coordinates of the left and right edges in the vertical direction. PPs then interpolate colors for each pixel in horizontal direction. The HOT structure can reduce the power consumption in the following ways.

- A 2D configuration with distributed DRAMs reduces the capacitance of the communication bus by hierarchical isolation from the EP_bus and PP_bus.

- Local connections allow frequent data access so that physically short-distanced wires can be used.
- Local connections also eliminate the use of power-hungry crossbar switches.

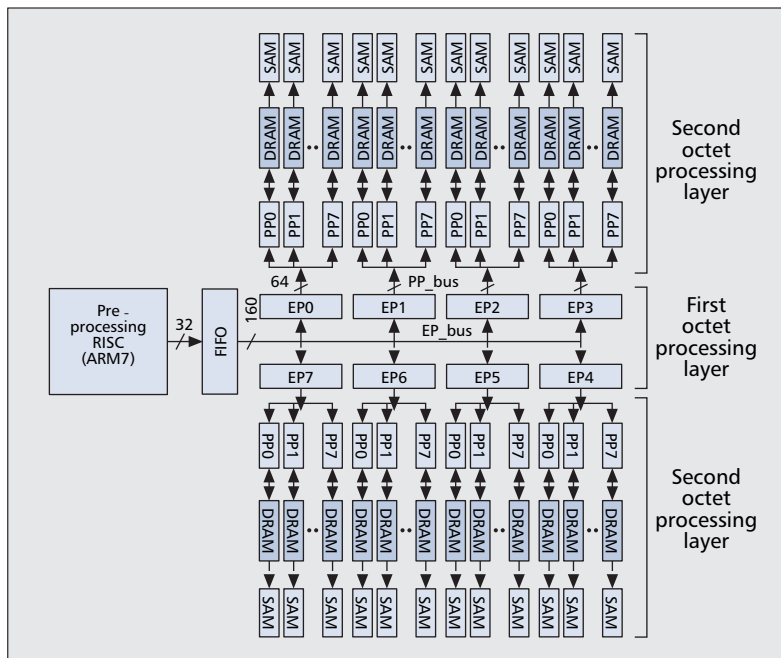
Additionally, partial wordline activation (PWA) is used to reduce the power consumption in the DRAM core.

A test chip containing one ARM7-compatible RISC, one EP, eight PPs, and eight DRAMs (a total of 512 kb) was fabricated with 0.35 μm three-metal Hynix merged-DRAM process. Its area and power consumption were 56 mm^2 and 590 mW, respectively. RAMP-I is the first implementation of mobile 3D graphics using embedded DRAM technology.

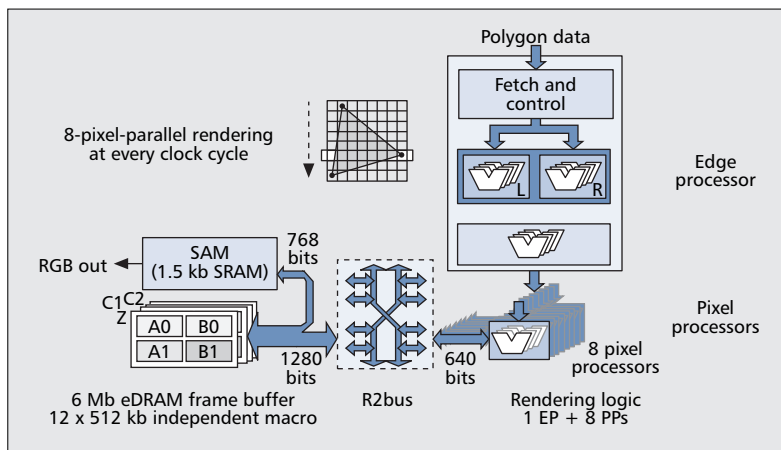
RAMP-II — RAMP-II integrates a 3D-rendering core, MPEG-4 decoding core, and embedded DRAM to provide complete multimedia solutions [4]. The rendering engine performs the same functions as RAMP-I with higher fill rate of 70 Mpixels/s, but only consumes 1/8 of the PPs. The proposed ViSTA (virtually spanning 2D array) structure of Fig. 3 consists of one EP, eight PPs, embedded DRAM, and R2bus (runtime reconfigurable bus). One EP calculates pixel data along the edges of an 8×8 clipped polygon and broadcasts them to eight PPs that can fill eight horizontal pixels simultaneously inside the polygon every clock cycle. Then EP calculates the next span-line repeatedly in a pipelined manner. The 12 embedded DRAM macros are placed together to form a *physically global* memory in order to enhance the cell efficiency. The DRAM is *logically localized* by the R2bus. Each PP operates as if it has its own local memory, as the R2bus changes the connection of the DRAM using a 2×2 crossbar switch and 8 to 16 bus shifters to assign the appropriate DRAM macro to the PPs for data read and write. Hence, the advantages of a local DRAM architecture are maintained: parallel access, high-memory bandwidth, and low power consumption. Memory power consumption is reduced using selective macro activation (SMA), partial wordline activation (PWA), and partial I/O activation (PIA).

RAMP-II contains an ARM9-compatible RISC processor with enhanced multiplier, MPEG4 accelerator, and 7.125 Mb embedded DRAM. The chip was fabricated with 0.18 μm six-metal Hynix merged-DRAM process. Its area and power consumption were 84 mm^2 and 160 mW, respectively. Merged-DRAM processes allow the use of 1T1C DRAM cells together with logic transistors, thus facilitating the implementation of high-performance memory-intensive applications. Since RAMP-II enables various multimedia such as MPEG4 video beyond real-time 3D graphics in a single chip, it is suitable for high-end smart-phones and PDA applications.

RAMP-IV — RAMP-IV [5] focuses more on real-time 3D gaming applications, drawing bilinear MIPMAP texture-mapped pixels with special rendering effects at 66 Mpixels/s and 264 Mtexels/s, as well as supporting the shading operations of previous RAMP architectures. Figure 4 shows the SlimShader architecture developed in

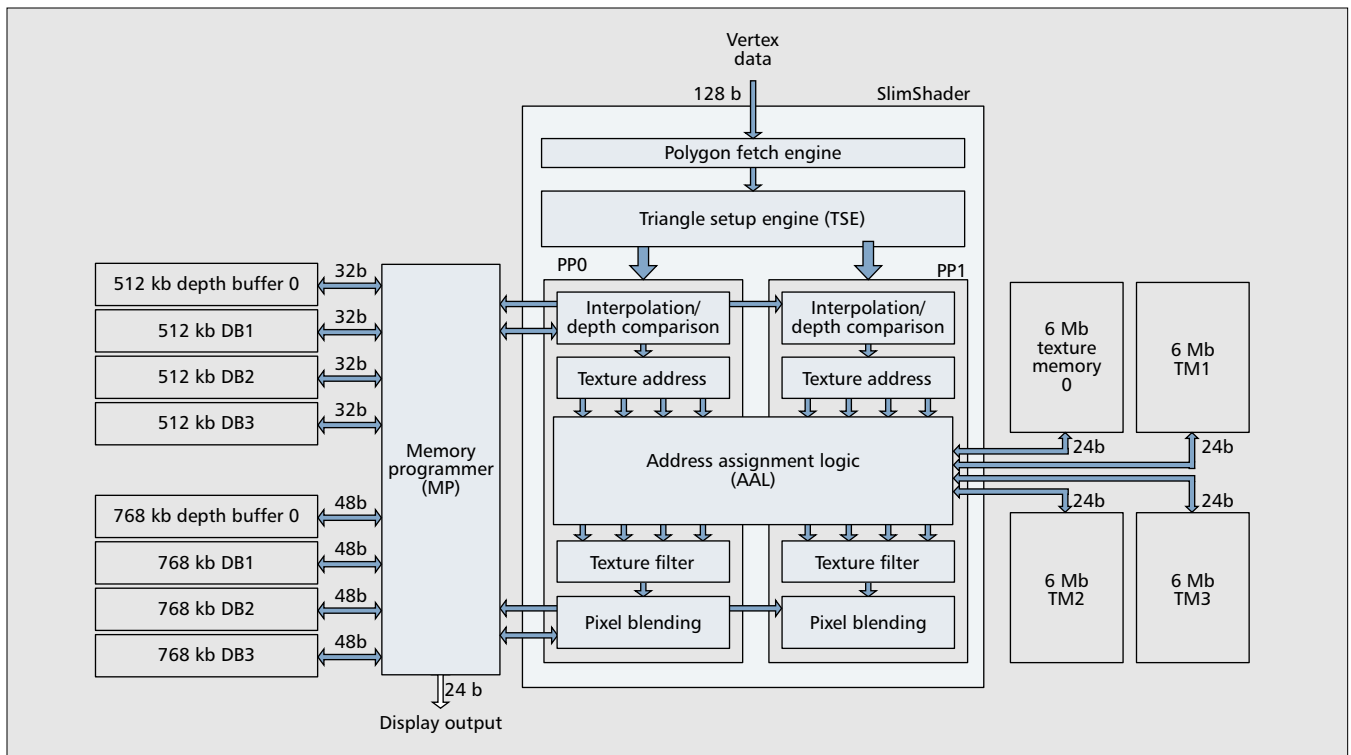


■ Figure 2. RAMP architectures: RAMP-I: HOT architecture.



■ Figure 3. RAMP architectures: RAMP-II: ViSTA architecture.

RAMP-IV. It consists of a triangle setup engine, an EP, two PPs, and 29 Mb of embedded DRAM. The reduced number of PPs is compensated by using a deeply pipelined PP structure, which enables a clock frequency of 50 MHz. Since texture mapping is a crucial function in real-time 3D graphics for creating realistic images, SlimShader contains two energy-efficient texture engines (TEs). In bilinear filtering, two pixels mapped to texel space require eight texture memory requests every cycle, causing huge power consumption. TEs employ address alignment logic (AAL), which uses temporal and spatial localities of texture addresses in MIPMAP-filtering to reduce total memory requests, thus yielding power saving. For real-time special effects such as fog, anti-aliasing, and cartoon shading, a memory programmer is implemented in SlimShader and post-processes the rendered pixels of frame buffer by using dedicated instruction set and single instruction multiple dath (SIMD) datapath.



■ Figure 4. RAMP architectures: RAMP-IV: SlimShader architecture.

RAMP-IV distributes the embedded DRAM over the logic pipeline via different ports, in addition to pixel-parallel distribution. Each pipeline stage can directly and concurrently access the contents of DRAM, just like accessing dedicated local SRAM. Satisfying the pipeline timing is a big challenge in terms of DRAM design as the cycle time (T_{RC}) of embedded DRAMs must be less than 20 ns, while commodity SDRAMs are working at 65 ns or more. The timing budget of frame and depth buffers is even stricter as the read-data must be written back to the same address within a single cycle for efficient read-modify-write (RMW) transactions. Distributing the DRAMs over the pipeline and accessing one or more of them selectively can reduce the power consumption of memory by 65 percent. Since the depth of the processed pixel is compared at the first stage of the PP pipeline, the following stages and corresponding memories can be gated off according to the comparison result.

The SlimShader architecture was integrated into a RAMP-IV chip together with an ARM9-compatible RISC processor with enhanced multiply-and-accumulate (MAC), 29 Mb of embedded DRAM, and a power-management unit. The chip was fabricated using a 0.16 μm Hynix 256 Mb SDRAM process. Its area and power consumption were 121 mm^2 and 210 mW, respectively. The RAMP-IV chip utilizes a pure DRAM process to reduce the fabrication cost. Although the pure DRAM process has slower logic transistor speed and fewer metal layers, 133 MHz speed can be achieved in the chip's RISC processor. The negligible subthreshold leakage current of the DRAM process also reduces standby current, which is a critical issue for battery-driven devices. Since modern SoC design prefers standard

CMOS logic process to DRAM-based CMOS process, the application of SlimShader architecture to a CMOS logic process is also being completed for future integration in next-generation graphics processors, while improving the scalability of memory capacity.

DIGIACC-I GRAPHICS PROCESSOR ARCHITECTURE

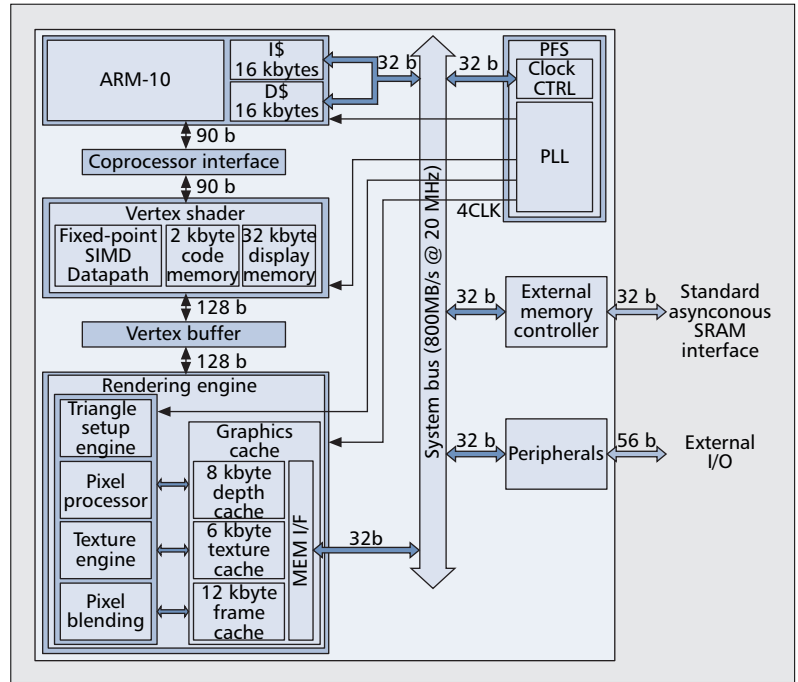
The current trend in mobile 3D graphics research is the development of high-quality flexible graphics architectures capable of providing more realistic images for handheld devices using advanced graphics algorithms. The DigiAcc-I processor [6] shown in Fig. 5 addresses the enhancement of the RAMP architecture with regards to high-performance 3D geometry and *programmability*. The DigiAcc-I graphics processor contains an ARM10-compatible 32-bit RISC processor with 16 kB I/D caches, a 128-bit programmable fixed-point SIMD vertex shader, a low-power rendering engine, and a programmable frequency synthesizer (PFS). DigiAcc-I features a programmable vertex shading architecture called SATINE. The SATINE vertex shader is implemented as an *ARM10 coprocessor* and processes all per-vertex operations such as geometry transformation and lighting calculation. Primitive assembly methods such as clipping and culling are also performed by the vertex shader in collaboration with the ARM10 processor. Figure 6 shows internal architecture of the SATINE vertex shader. The vertex program control unit (VPCTRL) issues the graphics instructions independently of the ARM10 processor for vertex shading. SATINE can also execute general-purpose integer and fixed-point

SIMD instructions controlled via the coprocessor interface in order to implement various multimedia operations beyond 3D graphics such as MPEG4 video decoding. A 32 kB display buffer integrated into the SATINE vertex shader decreases the system-bus bandwidth requirements when used in vertex array implementation and indexed primitive drawing. It stores vertex model data as well as graphics parameters such as matrix and light coefficients. The rendering engine employs the low-power 128-bit SlimShader architecture from RAMP-IV with a 26 kB dedicated graphics cache system (Fig. 7). Power consumption and performance analyses of various cache configurations yielded efficient cache characteristics of more than 90 percent hit ratio to cover the relatively small screen size in mobile terminals. The pixel data transfers are supported through the ARM system bus interface rather than by using the power-consuming memory crossbar. The PFS reduces the dynamic power consumption of the chip via software-controlled clock gating and frequency scaling of four independent clock domains. The frame rate of DigiAcc-I can be configured adaptively in runtime according to the required scene complexity. DigiAcc-I implements instruction-level clock gating of the SATINE vertex shader through the coprocessor interface, and pixel-level clock gating of the SlimShader rendering engine by depth-first graphics pipeline.

DUAL OPERATIONS

SATINE implements *dual operations* [7]. Unlike a conventional ARM coprocessor architecture, the SATINE vertex shader has dual operating states to allow it to be better adapted to the parallelism inherent in graphics processing.

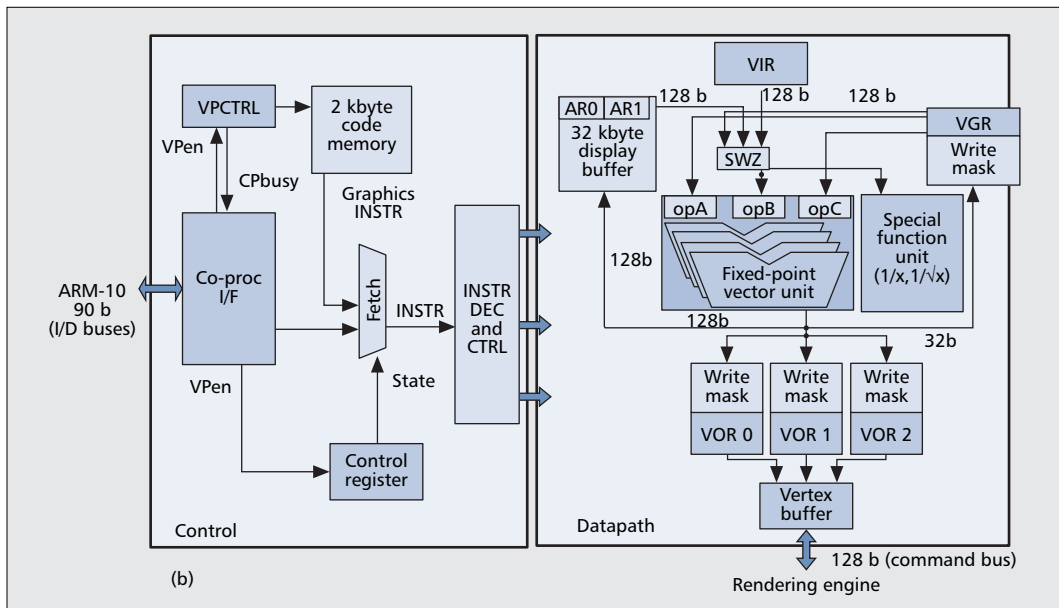
Tightly Coupled Coprocessor (TCC) State — In this state, SATINE is a normal ARM10 coprocessor. The instructions of the coprocessor are issued in the instruction stream of the main processor as extended coprocessor instructions, and they are executed in lock step with the



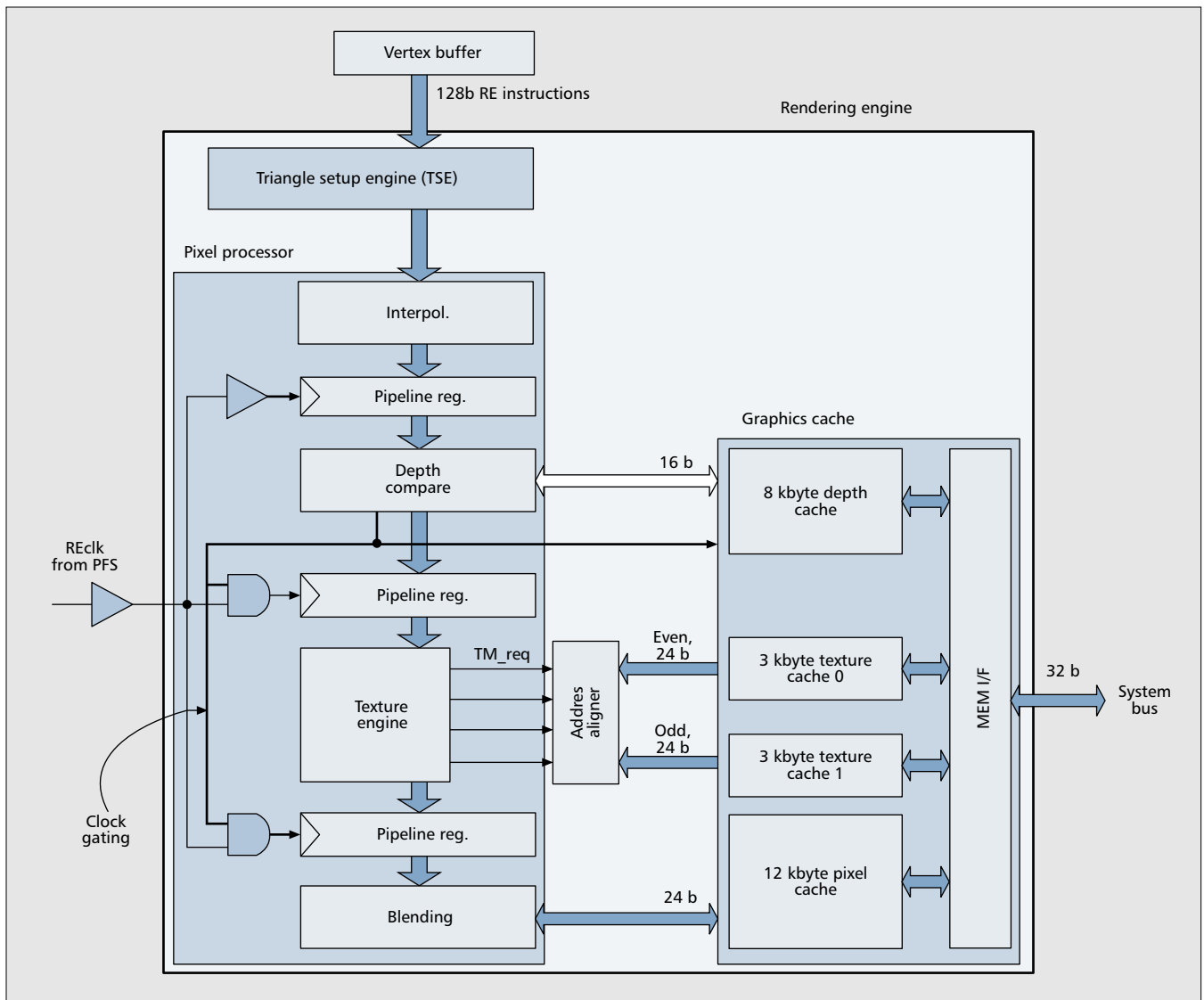
■ Figure 5. DigiAcc-I programmable graphics processor: a) overall architecture.

pipeline of the main processor. TCC state implements integer and fixed-point SIMD data processing instructions, and all instructions can be executed conditionally like conventional ARM instructions.

Parallel Processor (PP) State — In this state SATINE is an independent processor and can operate independently of the ARM10 processor. The PP state has a graphics instructions set that is separate from the general SIMD instructions of the TCC state. SATINE executes the independent vertex program while the ARM10 processor performs the main application program or enters a cache miss state. In the programmer's



■ Figure 6. DigiAcc-I programmable graphics processor: SATINE vertex shader.



■ **Figure 7.** *DigiAcc-I programmable graphics processor: rendering engine with graphics cache.*

model, the graphics instructions set is a subset of the general SIMD instructions with graphics extensions, such as source swizzling and write-masks. In the PP state, there are more register file sets that can be used as input operands of instructions. SATINE maintains the communication protocol of the ARM10 coprocessor interface by driving the coprocessor busy signal to the ARM10 processor, allowing the next coprocessor instruction from the ARM10 processor to be blocked for synchronization.

The two operating states share all of the hardware blocks except the instruction fetch units. Dual operations enable a single hardware resource to perform various multimedia operations. In addition, DigiAcc-I can process streaming graphics data more efficiently because various user-defined vertex processing can be performed for the current vertex input during the next vertex fetch of the ARM10 processor.

FIXED-POINT PROGRAMMABLE SHADING

Most 3D graphics systems require real-number representation to support various graphics algorithms. SATINE uses a fixed-point arithmetic

system in the instruction-set architecture and datapath design to reduce power consumption. Fixed-point datapaths use simple integer arithmetic circuits, so they consume less power and operate at higher clock frequencies than complex floating-point circuits while maintaining the same fidelity in 3D graphics tasks. In programmable vertex shading, prior knowledge of specific conditions of input vertex data can introduce more optimizations for cycle reductions such as the elimination of redundant matrix and vector calculations, as compared to fixed geometry operations. Moreover, the simple integer arithmetic circuit elements of the fixed-point datapath reduce the overhead of design complexity in the RISC-like SIMD vertex shader.

The accuracy of arithmetic results can be improved in mobile applications, since 32-bit fixed-point number systems contain more significant digits than 32-bit single precision floating-point numbers. However, dynamic range is limited in fixed-point systems, causing overflow and underflow in multiplication operations. The fixed-point datapath of SATINE can manipulate the fixed-point number ranging from Q32.0 to

Architecture	Hardware acceleration	3D feature	2D feature	Integration interface	Power consumption	Performance	Graphics index
PowerVR's MBX HR-S	Rendering + geometry (option)	Lighting, multitexturing, bump mapping	N/A	AMBA	208 mW @ 120 MHz	2.5 Mvertices/s 480 Mpixels/s	12.0 KVXPS/mW
Mitsubishi's Z3D		Texturing, multilight, shading	Rectangle fill, bit block transfer	Off-chip bus	38 mW @ 30 MHz	185 kvertices/s 5.1 Mpixels/s	4.9 KVXPS/mW
SONY's PSP	Geometry + rendering	Surface engine, vertex blending, multitexturing	H/W H.264 decode, AAC/MP3 audio codec	Standalone	500 mW @ 166 MHz	35 Mvertices/s 664 Mpixel/s	70 KVXPS/mW
ATI's Imageon2300		Texturing, vertex fog	MPEG4 decoder JPEG codec	N/A	N/A @ 100 MHz	1 Mvertices/s 100 Mpixels/s	N/A
nVidia's SC10	Rendering only	Shading, multitexturing	H/W MPEG4 codec H/W JPEG codec 64 b 2D engine	Off-chip bus	75 mW @ 72 MHz	1 Mvertices/s 72 Mpixels/s	13.3 KVXPS/mW
KAIST's DigiAcc-1	Geometry + rendering	Vertex programming, shading, texturing	General-purpose integer SIMD	ARM10 coprocessor	155 mW @ 200 MHz	50 Mvertices/s 50 Mpixels/s	161.2 KVXPS/mW
nVidia's GeForce 6800 (PC graphics)	Geometry + rendering	Rich vertex and pixel shading	N/A	AGP bus	< 10 0W	600 Mvertices/s	>6 KVXPS/mW

■ **Table 2.** Performance comparison.

Q1.31 in order to cover a wider dynamic range, while OpenGL-ES supports only the Q16.16 format. Hardware status registers are provided to indicate overflow and underflow conditions for enhanced error debugging.

The TCC state of SATINE implements two special instructions, controlled ADD/SUB (CAS) and controlled logical shift (CLS), for software floating-point emulations in an integer SIMD architecture. The CAS and CLS instructions change the cycle-consumed control-flow instruction sequences to single-cycle SIMD arithmetic operations, thus enhancing SIMD parallelism in floating-point calculations.

SYSTEM EVALUATION

The DigiAcc-I graphics processor (Fig. 8) was fabricated in 0.18 μm six-metal standard CMOS logic process. The processor's area is 36 mm^2 , including 2M logic transistors and 96 kB SRAM. The processor consumes 155 mW at 200 MHz, while drawing full geometry-processed, 24-bit true colored, and texture-mapped graphics images at a sustained 3.6 Mpolgons/s.

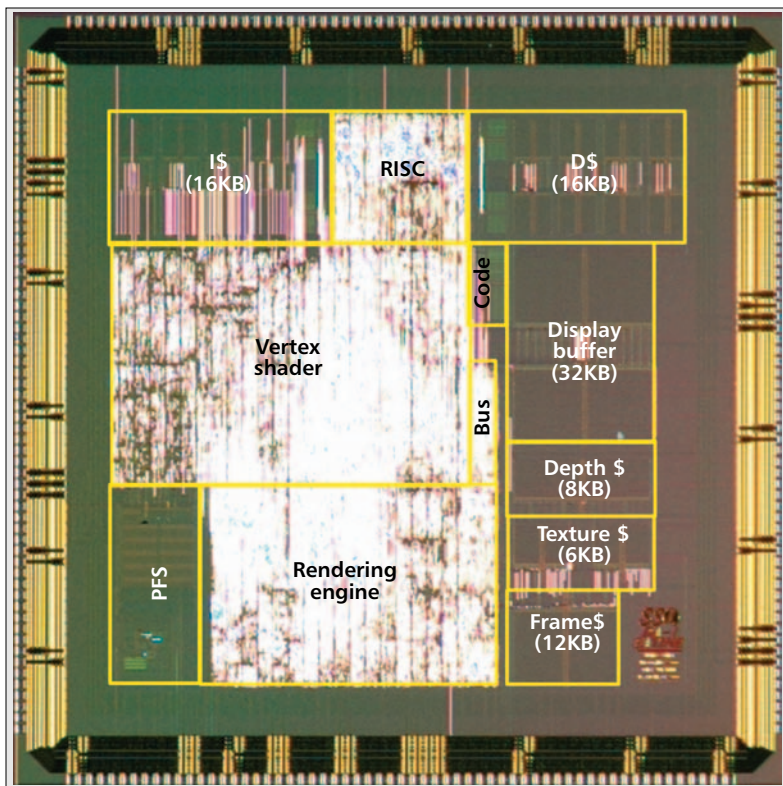
The REMY platform (Fig. 9) was developed to evaluate and demonstrate mobile 3D graphics using a flexible topology and protocol. The REMY platform incorporates Intel's PXA255 host system, since the prototype chip does not implement subsidiary hardware blocks, such as a memory management unit and an LCD controller. The host system is used for displaying and accessing the target system while varying the

configuration parameters, such as external memory capacity and bus protocols. The hardware layer of the REMY platform contains the target system, which is equipped with the fabricated chip and an FPGA system controller.

The mobile graphics library, MobileGL, was implemented in the software layer of the REMY platform to simplify development of applications. MobileGL is an OpenGL-ES compatible graphics library optimized with a handwritten assembly language so as to improve performance of an ARM-based mobile 3D graphics system. MobileGL consists of a fixed-point math library, vertex shader invocation routines, rendering engine-invocation routines, primitive assembly, and state variables with vertex-array capability. The native platform interface (NPI) provides the intrinsic functions of a hardware-dependent programmer's model during assembly and a high-level language for the core of the MobileGL. MobileGL can be ported to various hardware configurations without major architecture modifications by using NPI. The cycle-accurate software emulator of target hardware and the performance profiler were implemented in the REMY platform for performance evaluations and future derivative development.

PERFORMANCE COMPARISON

The graphics performance in mobile terminals cannot be compared directly in terms of processing speed such as vertex calculation rate because



■ Figure 8. System evaluation: DigiAcc-I chip.

the power consumption must be taken into account. For the comparison of the DigiAcc-I graphics processor with other previous architectures, the vertex-processing rate per second (VXPS) normalized by power consumption (mW) is used as the performance index. Table 2 summarizes the performance comparison and supported features of various graphics architectures. Based on the graphics index, KAIST's DigiAcc-I shows 161.2 kVXPS/mW, which is significantly higher than the result for other implementations.

Energy consumption is proportional to the amount of memory access, so many researchers focus on reducing off-chip bandwidth to enhance the battery lifetime for mobile 3D applications. PowerVR's MBX architecture reduces the memory accesses with tile-based rendering, but the performance is limited by the system bus and tiling overhead. Mitsubishi's Z3D core, intended for mobile phones, utilizes clock gating to achieve the lowest power consumption in spite of a floating-point geometry engine and 1 Mbit embedded SRAM [8]. However, its performance and functionality are constrained by the low operating frequency for limited power budget. The Playstation Portable (PSP) [9], developed by SONY, contains all necessary hardware blocks required for various applications in a handheld video game system, including a MIPS processor with vector FPU, media processing unit, rendering engine, and surface engine. The PSP features 2 Mb of embedded DRAM to boost internal memory bandwidth and support RMW operations for 3D graphics. The rendering engine and surface engine can execute more advanced graphics

algorithm such as tessellation, skinning, and morphing. The PSP also enables H.264 decoding for mobile video applications. However, the relatively high power consumption of the PSP limits its application in mobile terminals such as cell phones. nVidia's SC10 provides complete hardware acceleration for mobile multimedia [10]. It supports 2D/3D graphics and MPEG4 video with camera functions. The SC10 distinguishes itself from other architectures by implementing pixel-level programmability, such as blending and combining operations, for more realistic graphics images on handheld displays. However, the SC10 lacks a geometry engine for balanced performance.

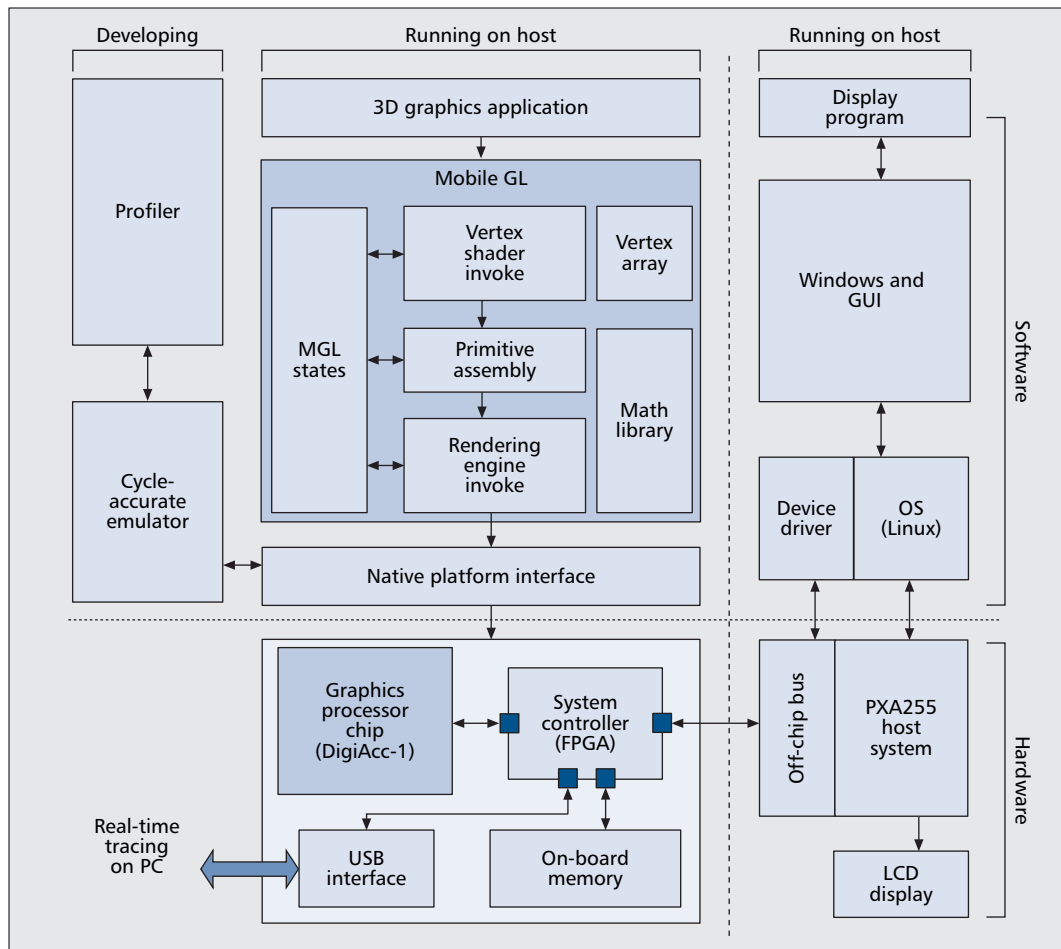
The design considerations in RAMP and DigiAcc demonstrate the high level of energy efficiency that is achievable by scaling and optimizing a processor's graphics functionality. The main design focus is on a simple programmable architecture optimized for mobile platforms, such as ARM processors, while achieving high performance with low power consumption.

CONCLUSION

The RAMP and DigiAcc architectures have been proposed in order to realize 3D graphics in low-power mobile terminals. Four architecture variants were developed and implemented, and their performance, power consumption, chip size, and cost were evaluated. DigiAcc-I, the programmable graphics processor for mobile terminals, was demonstrated on the REMY development platform, working with MobileGL, an OpenGL-ES compatible graphics library. Currently, we are working on accelerating data streaming with pixel-level programmability for low-power devices in order to balance the 3D pipeline and boost overall performance. Future research will focus on the design of a fragment shader and scalable data stream engine for generating more photo-realistic pixels with high sustained throughput.

REFERENCES

- [1] J.-H. Sohn *et al.*, "Optimization of Portable System Architecture for Real-time 3D Graphics," *Proceedings of IEEE Int'l. Symp. Circuits Syst.*, 2002, pp. 1769–72.
- [2] S.-J. Park *et al.*, "A Reconfigurable Multilevel Parallel Texture Cache Memory with 7.5GB/s Parallel Cache Replacement Bandwidth," *IEEE J. Solid-State Circuits*, vol. 37, no. 5, 2002, pp. 612–23.
- [3] Y.-H. Park *et al.*, "A 7.1GB/s Low Power Rendering Engine in 2D Array Embedded Memory Logic CMOS for Portable Multimedia System," *IEEE J. Solid-State Circuits*, vol. 36, no. 6, 2001, pp. 944–55.
- [4] C.-W. Yoon *et al.*, "A 80/20MHz 160mW Multimedia Processor Integrated with Embedded DRAM, MPEG4 and 3D Rendering Engine for Mobile Applications," *IEEE J. Solid-State Circuits*, vol. 36, no. 11, 2001, pp. 1758–67.
- [5] R. Woo *et al.*, "A 210mW Graphics LSI Implementing Full 3D Pipeline with 264Mtexels/s Texturing for Mobile Multimedia Applications," *IEEE J. Solid-State Circuits*, vol. 39, no. 2, 2004, pp. 358–67.
- [6] J.-H. Sohn *et al.*, "A 50Mvertices/s Graphics Processor with Fixed-point Programmable Vertex Shader for Mobile Applications," *IEEE Int'l. Solid-State Circuits Conf.*, 2005, pp. 192–93.
- [7] J.-H. Sohn *et al.*, "A Programmable Vertex Shader with Fixed-point SIMD Datapath for Low Power Wireless Applications," *Proc. ACM Graphics Hardware Wksp.*, 2004, pp. 107–14.



The RAMP and DigiAcc architectures have been proposed in order to realize 3D graphics in low-power mobile terminals. Future research will focus on the design of a fragment shader and scalable data stream engine for generating more photo-realistic pixels with high sustained throughput.

■ Figure 9. System evaluation: REMY platform.

[8] M. Kameyama *et al.*, "3D Graphics LSI Core for Mobile Phone: Z3D," *Proc. ACM Graphics Hardware Wksp.*, 2003, pp. 60–67.
 [9] M. Okabe *et al.*, "A 90nm Embedded DRAM Single Chip LSI with a 3D Graphics, H.264 Codec Engine, and a Reconfigurable Processor," *HotChips*, vol. 16, 2004.
 [10] E. Hutchins *et al.*, "SC10: A Video Processor and Pixel Shading GPU For Handheld Devices," *HotChips*, vol. 16, 2004.

BIOGRAPHIES

JU-HO SOHN (sohnuho@eeinfo.kaist.ac.kr) received B.S. and M.S. degrees in electrical engineering from the Korea Advanced Institute of Science and Technology (KAIST), Korea, in 2001 and 2003, respectively. He is currently working toward a Ph.D. degree in the same department at KAIST. His research interests include low-power high-performance circuits and multimedia system design with specific interest in 3D computer graphics architecture and its implementation for mobile applications.

YONG-HA PARK (yongha.park@samsung.com) was graduated from the electronics department of Kyungpook National University, Taegu, Korea, in 1996 and received M.S. and Ph.D. degrees in electrical engineering from the Korea Advanced Institute of Science and Technology (KAIST), Taejeon, in 1998 and 2002, respectively. His Ph.D. work concerned application-specific embedded memory logic focusing on single-chip 3D graphics rendering. In 2002, he joined Samsung Electronics Co., Ltd., Kyungkido, Korea, working on design of application processors (AP) for mobile handsets. He also designs multimedia intellectual property (IPs) for video and 3D graphics applications of AP products.

CHI-WEON YOON (chiweon.yoon@samsung.com) received B.S., M.S. and Ph.D. degrees in electrical engineering from the Korea Advanced Institute of Science and Technology

(KAIST), Taejeon, Korea, in 1997, 1999, and 2004, respectively. His research interests include application-specific embedded memory logic design, and low-power VLSI architecture for video processing and 3D graphics. Currently, he is with Samsung Electronics, where he designs high-speed and high-density flash memories.

RAMCHAN WOO (ramchan@ti.com) received B.S. (summa cum laude), M.S. and Ph.D. degrees in electrical engineering from Korea Advanced Institute of Science and Technology (KAIST), in 1999, 2001, and 2004, respectively. At KAIST, he developed low-power 3D graphics SoCs integrated with embedded DRAM for mobile applications. Since 2004, he has been with Texas Instruments, Dallas, TX, where he is a system architect of multimedia chips for camera phones in the Imaging and Audio Group of the DSPS division. His research interests include low-power and low-cost integration of imaging, video, and graphics pipelines into single SoC for wireless terminals.

SE-JEONG PARK (sjpark@core.kaist.ac.kr) received a Ph.D. degree from Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea, 2002. He worked for IDIS Co. Ltd. as a principal engineer from 2002 to 2004. From April 2004, He is working for KAIST as an invited research professor. He majored in 3D graphics architecture and its low-power SoC implementation using merged DRAM logic (MDL) process technology.

HOI-JUN YOO [M] (hjyoo@ee.kaist.ac.kr) received MS and Ph.D. degrees in electrical engineering from KAIST in 1985 and 1988, respectively. He is with the Department of Electrical Engineering faculty at KAIST, where he leads the System Integration and IP Authoring Center. His research interests include SoC design, IP authoring, high-speed and low-power memory circuits and architectures, design of mobile multimedia system, and novel devices and circuits. He was the SoC project manager of the Korea Ministry of Information and Communication from 2003 to 2005.