A 1.2Mpixels/s/mW 3-D Rendering Processor
For Portable Multimedia Application

Jeong-Ho Woo, Min-Wuk Lee, Hyejung Kim, Ju-Ho Sohn and Hoi-Jun Yoo
Semiconductor System Laboratory, Department of EECS
Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea
denber@eeinfo.kaist.ac.kr, hjyoo@ee.kaist.ac.kr

Abstract — A 1.2Mpixels/s/mW 3-D rendering processor is designed and implemented for portable multimedia application. A sustained 20Mpixels/s pixel fill rate is obtained at 10MHz with the help of embedded depth buffer and recursive sub-block mapped texture cache. In order to achieve low power rendering operation, SlimShader architecture with non-atomic read-modify-write scheme and logarithmic datapath is employed. The rendering processor consists of 281k logic gates and 164kB embedded SRAM in 25mm². It is fabricated in 0.18µm 1-poly 6-metal CMOS logic process and real-time 3-D graphics images are successfully demonstrated on the system evaluation board. It consumes 17mW at 10MHz while drawing texture-mapped 3-D graphics images.

I. INTRODUCTION

The real-time 3-D computer graphics are one of the most important applications in portable multimedia devices such as 3G cell-phones or PDAs [1, 2, 3, 10]. To draw realistic 3-D graphics images, texture mapping is widely used. Drawing texture-mapped 3-D graphics images on mobile terminals requires huge memory bandwidth and capacity for storing the color data, depth data and texture images. How to provide the effective memory bandwidth for efficient rendering is one of the most important design issues in portable graphics system.

To get enough memory bandwidth, using the embedded DRAM (eDRAM) technology is one candidate solution [1]. Though eDRAM provides huge memory bandwidth, its process technology is still very expensive and yield is not optimized yet.

3-D rendering graphics functional blocks were developed as Intellectual Properties (IP) for SoC [2, 3]. Since the data used for graphics processing has high data locality, these IPs used graphics caches. However, the optimal cache configurations and mapping schemes for low power consumption have not been examined yet.

In this work, we design and implement a low power 3-D rendering processor focused on high sustained performance and low power consumption. The proposed processor enables high speed drawing, 2pixels/cycle, by embedded depth buffer and recursive sub-block (RSB) mapped texture cache. For low power consumption, it exploits 3 techniques, non-atomic read-modify-write (NARMW) scheme, logarithmic datapath [6] and clock gating. The details of the high speed schemes and low power techniques will be explained in the following sections.

II. SYSTEM ARCHITECTURE

Recently, embedded RISC processors such as Xscale provide 400MB/s system bus bandwidth but in reality, available bandwidth is limited to less than 50% due to bus contention by multiple IPs [4]. The drawing of the texture mapped image at 20Mpixels/s requires 360MB/s memory bandwidth for transferring graphics data such as depth data, color data or texture data [Fig 1(b)]. However, the bandwidth of the system bus is allocated as shown Fig 1(a) during the operation of 3-D graphics. According to Fig 1(a), the available memory bandwidth for graphics data transfer is less than 45MB/s. This limited bandwidth deteriorates the rendering performance to 1.25MB/s, which is much below the minimum system requirement, 3Mpixels/sec. In this study, new system architecture is proposed to alleviate the bandwidth bottleneck.

A. System Architecture and Low Power Rendering Engine

Fig. 2 shows a portable system block diagram incorporating the proposed rendering processor. The host processor performs geometry operation using primitive data stored in system memory. The vertex data and rendering
commands are transferred to rendering processor through
system bus. The rendering processor generates texture-
mapped pixels which are displayed on LCD of QVGA
screen resolution.

In order to reduce data transfers between the rendering
engine and system memory, separated data bus and SRAM
are used for graphics data. Of course, the SRAM can be
accessed by host processor by direct access mode. In direct
access mode, the processor stores texture images at SRAM
or read frame buffer from SRAM. Usually, host processor
accesses the rendering engine to transfer vertex data and
commands.

Proposed 3-D rendering processor consists of low power
rendering engine (LRE), depth buffer, texture cache and
interface logics as shown in Fig. 3.

The system interface logic decodes memory address and
decides operation modes between graphic mode and direct
memory access mode. The memory interface logic controls
the SRAM, and arbitrates memory requests from texture
cache and blending unit.

LRE performs shading and texturing operations with two
pixel processors including texture units. It employs the
SlimShader architecture [1]. In order to achieve low power
rendering operation, it uses pixel-level clock gating and
address alignment logic. Pixel-level clock-gating prevent
unnecessary operation of pipeline datapaths according to
the results of depth comparison. The pixel-level clock gating
shows average of 25% power reduction for typical graphics
applications. Bilinear MIPMAP texture filtering requires as
many as 8 texture memory requests at every cycle. The
depth address alignment logic reduces memory requests from 8 to
average of 2.5 by using temporal and spatial data locality.

B. Logarithmic texture datapath

In order to implement perspective correct texture
mapping, per pixel division is required which can be
described as the following equation.

\[ U = \frac{u}{w} \quad \text{and} \quad V = \frac{v}{w} \quad - (1) \]

(Where \( 0 \leq (U, V) \leq 1 \))

Since the embedded 3-D graphics system such as
OpenGL|ES requires 32bit texture coordinate [8], a 32bit by
32bit divider is required for embedded 3-D graphics.
However, general divider such as RADIX-4 requires several
cycles to calculate 32bit division, resulting in the rendering
performance degradation.

To implement low latency 32bit divider with low power
consumption, a logarithmic divider [6] is used. By
implementing logarithmic divider, 32bit by 32bit division
can be done in a single cycle with 3mW power consumption.
The use of logarithmic dividers in texture unit reduces the
power consumption of LRE by 1.2mW compared to a
previous work [3].

III. LOW POWER MEMORY ARCHITECTURE

The proposed memory architecture consists of a 160kB
embedded depth buffer and 4-RSB mapped texture caches as
shown in Fig. 4. In depth buffer, the NARMW scheme and
partial activation scheme are used to reduce its power
consumption. In addition, RSB mapping is adopted to reduce
power consumption of the texture cache.

A. Embedded Depth Buffer and partial activation

Since depth comparison is frequently performed, latency
of the depth buffer should be minimized. Conventionally,
multi-level depth cache is used in PC graphics but it
consumes high power and thus is not suitable for portable 3-
D graphics [9].

To reduce latency for portable 3-D graphics, we integrate
160kB full depth buffer. Since only two depth requests exist
in every rendering cycle, only two memory-address spaces
need to be active while other memory spaces go to idle mode
to reduce power consumption.

In this work, the depth buffer consists of 2 buffers, odd
buffer and even buffer. And each buffer is partitioned into 5

![](image1.png)

**Figure 2. Block Diagram of a Portable System**

![](image2.png)

**Figure 3. Block Diagram of Rendering Processor**

![](image3.png)

**Figure 4. Block Diagram of Memory Architecture**
Therefore, the memory clock can be lowered to twice of the required data only by read and write operations. Windows for the memory access so that memory can provide NARMW is pipelining of read, modify and write cycles performance degradation, we use NARMW. The key idea of clock frequency is not suitable for achieving low power. eDRAM is not optimized for SoC design, and increasing 4 times higher than the rendering clock has been used to RMW operation. eDRAM [1] or memory clock of at least 3-
during RMW, it is necessary to implement a single cycle rendering performance heavily depends on memory latency for depth-comparison and pixel alpha blending. Since 3-D memory access pattern in typical 3-D graphics applications B.

Non-Atomic Read-Modify-Write

Read-Modify-Write (RMW) is the most frequent memory access pattern in typical 3-D graphics applications for depth-comparison and pixel alpha blending. Since 3-D rendering performance heavily depends on memory latency during RMW, it is necessary to implement a single cycle RMW operation. eDRAM [1] or memory clock of at least 3-
time. Therefore, the memory clock can be lowered to twice of the rendering clock frequency to achieve the single cycle RMW.

However, a problem may arise due to the read-after-write (RAW) data hazard. If a 3-D application continuously draws a pixel at the same position, the read and the write address will be the same, which will lead to the RAW hazard. This can be solved by data forward buffer in the NARMW controller, as shown in Fig. 6(b).

NARMW controller consists of input latches, forward detector, forward buffer and PA generator. The forward detector compares read address with write address, and decides whether the depth data goes to LRE or not. When the data hazard is occurred, the forward detector transfers the data stored in forward buffer to LRE because depth buffer does not contain correct data. In other case, the forward detector transfers data from depth buffer to LRE. The PA generator decodes LRE’s depth buffer address and enables memory module to control PADB.

By using NARMW, the power consumption of the depth buffer is reduced by 33% because the memory clock frequency is lowered from 3x to 2x of LRE frequency.

C. Recursive Sub-Block Mapped Texture Cache

In the previous work [10], tiled-order rasterization was employed for portable 3-D graphics. However, it requires high computation power because the object data for the whole scene must be tiled before rasterization. Moreover, it requires high design complexity. Therefore, in this work, we use horizontal order rasterization for low power consumption. But, the horizontal order rasterization increases conflict miss during texture mapping when conventional block mapped [7] texture cache is used [Fig 7(a)].

In order to reduce conflict miss, we propose RSB mapping. In the RSB mapping, 256x256 texture image is divided into four 64x64 blocks and the 64x64 block is divided into four 16x16 sub-blocks. The block is divided recursively until sub-block becomes 2x2. Then sub-blocks...
are stored successively in memory as a tiled fashion shown in Fig. 7(b). By changing image order, the RSB mapping representation induces the balanced storing of texture data, which reduces conflict miss to 4 independent of access directions. Fig. 8 shows the simulation results of RSB mapping. In a typical 3-D application, the RSB mapped texture cache shows average of 0.4% miss rate and the power consumption is reduced by 17.4% compared to 8x8 block mapped texture cache [5].

IV. IMPLEMENTATION AND RESULTS

The proposed 3-D rendering processor is fabricated using 0.18μm CMOS logic process. It consists of 281k logic gates and 164kB embedded SRAM. Fig. 9(a) shows the die photograph and Table I summarizes its features. It can draw 16bit mapped pixels with the rendering speed of 20Mpixels/sec at 10MHz consuming less than 17mW. The use of NARMW scheme and PADB reduces power consumption of depth buffer to 5.2mW. The power consumption of texture cache is reduced by 17.3% with the help of RSB mapped texture cache and it consumes 3mW. The logarithmic dividers in texture unit reduce power consumption of LRE by 1.2mW and in result the rendering engine consumes only 8.8mW. Table II shows the performance comparison of this work and some previous works. The power consumptions are normalized to 1.8V supply voltage. Since the power consumption is important as much as fill rate in a portable system, the performance is obtained using fill rate normalized to power consumption.

To verify the operation of implemented rendering processor, a system evaluation board is designed with Intel Xscale processor targeting multimedia PDA platform. Real-time 3-D graphics images with 16bit texture mapped pixels are demonstrated on the system evaluation board as shown in Fig. 9(b).

V. CONCLUSIONS

A 1.2Mpixels/s/mW 3-D rendering processor is designed and implemented for portable multimedia application. A sustained 2pixels/cycle rendering speed is achieved with the help of embedded depth buffer and texture cache. Applying various low power techniques such as NARMW scheme, PADB, RSB mapping and logarithmic divider reduces the power consumption to 17mW while keeping drawing texture-mapped real-time 3-D images. The rendering processor is fabricated using 0.18μm 1-poly, 6-metal CMOS logic process. It contains 281k logic gates and 164kB embedded SRAM macros in 25mm². Real-time 3-D graphics images are successfully demonstrated on the system evaluation board.

VI. REFERENCES