A 195mW, 9.1MVertices/s Fully Programmable 3D Graphics Processor for Low Power Mobile Devices

Jeong-Ho Woo, Ju-Ho Sohn, Hyejung Kim, Jongcheol Jeong¹, Euljoo Jeong¹, Suk-Joong Lee¹ and Hoi-Jun Yoo

Dept. EECS, KAIST, 373-1, Guseong-dong, Yuseong-gu, Daejeon, 373-1, Daejeon, KOREA
1 Corelogic, Inc., 6th FL., City Air Tower, 159-9, Samsung-dong, Gangnam-gu, Seoul, 135-973, KOREA
denber@einfo.kaist.ac.kr

Abstract—A 195mW, 9.1Mvertices/s fully programmable 3D graphics engine is designed and implemented in 0.13μm CMOS process for mobile devices. The power-optimized unified shader architecture provides high performance programmable vertex shading and programmable pixel shading with 35% area and 28% power reduction. The logarithmic lighting engine and specialized lighting instruction improves the vertex throughput to 9.1Mvertices/s including full OpenGL lighting, which is 2.5 times higher performance compared with previous works. The 3D graphics engine was successfully demonstrated on the system evaluation board.

I. INTRODUCTION

Recently, the 3D computer graphics is widely used for mobile consumer electronics such as the cellular phones and personal multimedia players (PMPs), for entertainment applications such as games or avatars. And the personal navigation applications require 3D graphics for realistic map display and convenient services.

In mobile devices, since users often hold the small screens closer to their eyes than PC, the average eye-to-pixel angle is larger than that of a PC [1]. Therefore, every pixel in mobile applications should be drawn with higher quality, which can be obtained with programmable vertex shading and pixel shading. However, until recently, previous researches did not provide a fully programmable graphics pipeline, which is required for high quality of graphics images because of the lack of power-optimized fully programmable 3D graphics architecture for mobile devices. They only integrate programmable vertex shader [3].

Moreover, the way to provide fully programmable 3D graphics of PC graphics is not suitable for mobile devices. The PC graphics uses several vertex shaders and pixel shaders but, they require huge power consumption and huge silicon area while mobile devices require low power consumption and small area. And the unified shader architecture [6] of the console devices, which merges vertex shader and pixel shader in a single hardware, can provide fully programmable 3D graphics in a single hardware but, it is not optimized for low power consumption and high graphics performance.

In this work, a low power fully programmable 3D graphics engine is presented for mobile devices. The power-optimized unified shader architecture is employed to provide high performance fully programmable 3D graphics pipeline with low power consumption and small area. To improve the vertex throughput, the dedicated lighting engine and specialized lighting instruction are employed. The lighting engine adopted logarithmic number system [4] to calculate complex lighting equation with high throughput and low power consumption. To reduce power consumption further, partial activation and clock gating schemes are employed.

II. FULLY PROGRAMMABLE 3D GRAPHICS ENGINE

A. Internal Architecture

Figure 1 shows a block diagram of the fully programmable 3D graphics engine (3DGE). It consists of Unified Shader (USH), Vertex Generator (VG), Fragment Generator (FG), Pixel Generator (PG), Matrix/Quaternion-Vector Generator (MG) and graphics caches.

The USH can fully support the programmable 3D graphics API, OpenGL|ES 2.0 for mobile devices with small area and low power consumption because the vertex shading and pixel shading are performed in a single USH. With the USH in 3DGE, the silicon area and power consumption are reduced by 35% and 28%, respectively. The texture engine in USH performs texture address generation, texture fetch and filtering. The vertex generator, fragment generator and pixel generator are designed to perform 3D graphics operations such as clipping and shading with high throughput and low power consumption. To reduce the power consumption during rendering operation, the pixel-level clock-gating [2] is employed.

The MG is employed to enhance the graphics performance. The USH uses floating-point matrices, which take thousands of cycles to generate on embedded RISC [3] and this overhead limits the 3D graphics performance. To
enhance the speed of the matrix generation, the MG is designed to have 6 floating-point multipliers, 6 floating-point adders, a floating-point divider, a floating-point square-root block and a floating-point trigonometric function block.

B. Pixel-Vertex-Multi-Threading

Figure 2-(a) shows a data flow diagram of the programmable 3D graphics operation. In the programmable 3D graphics, the USH performs both programmable vertex shading and pixel shading. The input vertex is computed using vertex program, which includes user-defined transformation and lighting (T&L) operation. After T&L operation, the VG and FG generate interpolated pixels and the pixels are modified in the USH using user-defined pixel program. And then, the PG generates final pixels.

During programmable pixel operation, the texture is widely used to modify pixels. However, the texture cache miss degrades 3D graphics performance because the USH wastes tens of cycles without any operation until texture cache to be filled. To enhance 3D graphics performance, the USH adopts a Pixel-Vertex-Multi-Threading (PVMT), which enables to utilize datapaths in parallel.

Since the texture engine of USH is independent of SIMD datapath and special function unit (SFU), the PVMT enables to calculate new vertices on SIMD and SFU while texture engine is halting. When the texture cache miss is occurred during programmable pixel operation, PVMT changes the USH mode to programmable vertex operation and a new vertex is issued on USH. After finishing the programmable vertex operation, the USH move back to the programmable pixel operation and finish the programmable pixel operation.

As a result, with the help of PVMT, pixels and vertices are computed on single USH at the same time as shown in Figure 2-(b) and it improves 3D graphics performance in real 3D graphics applications.

III. Unified Shader Architecture (USH)

A. Internal Architecture

USH is a SIMD processor for programmable 3D graphics. It consists of 128b, 4x32bit SIMD datapath, special function unit (SFU), texture engine, specialized lighting engine (LE), dedicated register file and control logic as shown in Figure 3.

The SIMD datapath is responsible to vector arithmetic operations such as addition and multiplication and the SFU calculates logarithm (LOG), exponent (EXP), reciprocal (RCP) and reciprocal square root (RSQ) in only 2cycles by using logarithmic number system (LNS) [4]. Since the vertex shading is performed in IEEE-754 floating point number system and pixel shading is performed in 32bit fixed-point number system, the SIMD and SFU are designed to handle both number systems in a single hardware.

For streaming graphics processing, USH contains multiple register files—input registers (IR), output registers (OR) and temporary SIMD registers (TR). The IR, used to hold the vertex attributes such as position and normal vector and pixel attributes such as position, color and texture coordinate. In order to reduce data fetch time, the IR consists of two register banks for double buffering. The TR is used to store temporary results during vertex program and pixel
program execution. The modified vertex and modified pixel information are transformed into OR.

B. Logarithmic Lighting Engine (LE)

Since the lighting calculation is the most complex calculation during programmable vertex operation, the LE is employed to improve vertex throughput with low power consumption. Figure 4 – (a) presents an example of OpenGL lighting equation including an ambient light, a diffuse light and a specular light. In the LNS, the POW operation of specular light is described with MUL and ADD as shown in Figure 4 – (a) and it can be easily computed with low power consumption. Therefore the LE has LNS datapath for the specular light and ordinary datapath for the ambient and diffuse light as shown in Figure 4 - (b).

Since, in the previous works, the data dependency in the lighting calculation degrades graphics performance as shown in Figure 4-(c), the specialized lighting instruction, TLT, is proposed to reduce data dependency. The TLT instruction combines the light coefficient calculation with the multiplication of coefficients and materials together and it generate lit-vertex in every two cycles with logarithmic LE. Figure 4-(c) presents the comparison of instruction sequences for OpenGL light calculation between proposed LE and previous implementation. By adopting logarithmic LE and TLT instruction, the USH calculates a lit-vertex in every two cycles and it achieves 9.1Mvertices/sec, which is 2.5 times higher performance compared with previous implementation [3].

IV. IMPLEMENTATION RESULTS

The developed 3D graphics engine is fabricated by a 0.13μm 7-metal CMOS logic process and it was integrated into ARM-9 based mobile multimedia SoC [5]. The 3D graphics engine contains 1.3M logic gates and graphics cache in 3.3mm x 3.0mm.

During 3D graphics application, the USH, logarithmic datapaths and low power schemes such as partial activation
and pixel-level clock gating [2] reduce the power consumption and the fully programmable 3D graphics engine consumes less than 195mW at 100MHz operating frequency and 1.2V supply voltage. The Table 1 summarizes features of the 3D graphics engine and Figure 5 shows the chip micro-photograph and the evaluation system, which targets multimedia cellular phone system.

Figure 6 shows the 3D graphics performance comparison with that of the previous implementations [2, 3]. The developed 3D graphics engine provides the fully programmable 3D graphics with low power consumption for mobile devices. With the help of TLT instruction and the logarithmic LE, the 3D graphics engine improves vertex fill rate by 2.5 times compared with the previous works and it achieves 2.1 times improvement in mobile graphics performance [3], which normalizes the 3D graphics performance by power consumption compared with previous implementation, respectively.

V. CONCLUSIONS

A low power fully programmable 3D graphics engine is designed and implemented for mobile devices. The power-optimized unified shader provides fully programmable 3D graphics pipeline with 35% area reduction and 28% power reduction. Logarithmic lighting engine and specialized lighting instruction improves 3D graphics performance to 9.1MVertices/s vertex fill rate, which is 2.5 times higher performance compared with previous works and PVMT scheme enhances 3D graphics performance in real applications. The developed 3D graphics engine consumes less than 195mW at 1.2V supply voltage and 100MHz operating frequency. The unified shader and logarithmic datapaths reduce the silicon area and the 3D graphics engine is implemented on 3.3mm x 3.0mm in 0.13 μm CMOS logic process.

VI. REFERENCES


<table>
<thead>
<tr>
<th>Process Technology</th>
<th>0.13μm 7M CMOS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power Supply</td>
<td>1.2V</td>
</tr>
<tr>
<td>Transistor Counts</td>
<td>1.3M logic gates</td>
</tr>
<tr>
<td>Die Size</td>
<td>3.3mm x 3.0mm</td>
</tr>
<tr>
<td>Operating Frequency</td>
<td>100MHz</td>
</tr>
<tr>
<td>Power Consumption</td>
<td>195mW@Full 3D Graphics Processing</td>
</tr>
<tr>
<td>Package</td>
<td>144pin CPGA</td>
</tr>
<tr>
<td>Geometry Performance</td>
<td>9.1Mvertices/s (including Full OpenGL Lighting)</td>
</tr>
<tr>
<td>Rendering Performance</td>
<td>100Mpixels/s, 450Mpixels/s</td>
</tr>
<tr>
<td>Programmability</td>
<td>Programmable Vertex Shader</td>
</tr>
<tr>
<td>Standard API</td>
<td>OpenGL 3.0, OpenGL 2.0</td>
</tr>
<tr>
<td>Screen Resolution</td>
<td>Up to 1024 x 1024 pixels</td>
</tr>
</tbody>
</table>

Table 1: Feature Summary

Figure 6. Performance Comparison