# A Low Power Multimedia SoC with Fully Programmable 3D Graphics and MPEG4/H.264/JPEG for Mobile Devices

Jeong-Ho Woo, Ju-Ho Sohn, Hyejung Kim, Jongcheol Jeong[1], Euljoo Jeong[1], Suk Joong Lee[1] and Hoi-Jun Yoo

Dept. EECS, KAIST, 373-1, Guseongdong, Yuseonggu, Daejeon, 305-701, KOREA

1 Corelogic, Inc., 6th FL., City Air Tower, 159-9, Samsungdong, Kangnumgu, Seoul, 135-973,KOREA

+82-42-869-8068

denber@eeinfo.kaist.ac.kr

## ABSTRACT

We present a low power multimedia SoC with fully programmable 3D graphics, MPEG4 codec, H.264 decoder and JPEG codec for mobile devices. The unified shader in 3D graphics engine provides fully programmable 3D graphics with 35% area and 28% power reduction. Logarithmic lighting engine and the specialized lighting instruction enable 9.1Mvertices/s vertex throughput. The merged JPEG/MPEG4 codec and the unified shader reduce the silicon area further and the SoC consumes 6.4mm x 6.4mm in 0.13µm CMOS logic process.

## Categories and Subject Descriptors

I.3.1. [**Computer Graphics**]: Hardware Architecture-Graphics Processors

## General Terms: Algorithms, Design, Performance

## Keywords

Mobile Multimedia SoC, Programmable 3-D Graphics, Low Power Design.

## 1. INTRODUCTION

Recently, multiple multimedia functions are merged into the mobile devices to be a personal multimedia terminal. The digital camera is widely incorporated and recently even digital multimedia broadcasting (DMB) and real-time 3D graphics are employed to provide entertainment applications [1]. In mobile devices, since users often hold the small screens closer to their eyes, the average eye-to-pixel angle is larger than that of a PC. Therefore, every pixel in mobile applications should be drawn with high quality, and a fully programmable 3D graphics engine is required [4].

Although there have been many publications on multimedia solutions [1-5], these chips did not integrate the full multimedia functions such as digital camera, video, audio and 3D graphics on a single die due to its huge gate counts and design complexity. Moreover, they could not provide a fully programmable 3D graphics, which is required for high quality images compatible with OpenGL|ES 2.0.

In this work, a low power multimedia SoC with full integration of a fully programmable 3D graphics and MPEG4, H.264 and JPEG processing is presented for mobile devices. Its 3D graphics engine is unique in that the unified shader (USH) [7] is used for fully programmable 3D graphics with low power and small silicon area, special lighting instruction and logarithmic [6] lighting engine are used for high 3D graphics performance. Its USH is power optimized single shader for mobile devices in contrast to that of console device which has multiple general USH.

This paper consists of six sections. The system architecture and video engine will be discussed in Section 2, and the programmable 3-D graphics engine and details of unified shader will follow in Section 3 and Section 4, respectively. The chip implementation of the SoC will be described in Section 5, and finally the conclusion of our work will be made in Section 6.

## 2. SYSTEM ARCHITECTURE AND VIDEO ENGINE

Most of the current mobile platforms employ the AMBA bus so that the ARM9 RISC processor, fully programmable 3D graphics engine, video engine, display engine and peripherals are connected to the AMBA bus as shown in Fig. 1.

The ARM9 RISC processor is used as a stand-alone host processor and the software audio codec is decoded on it. The display engine can display the rendered images by 3-D graphics engine or video engine to TV or LCD screen.

*ISLPED'07*, August 27–29, 2007, Portland, Oregon, USA.
Copyright 2007 ACM 978-1-59593-709-4/07/0008 …$5.00.

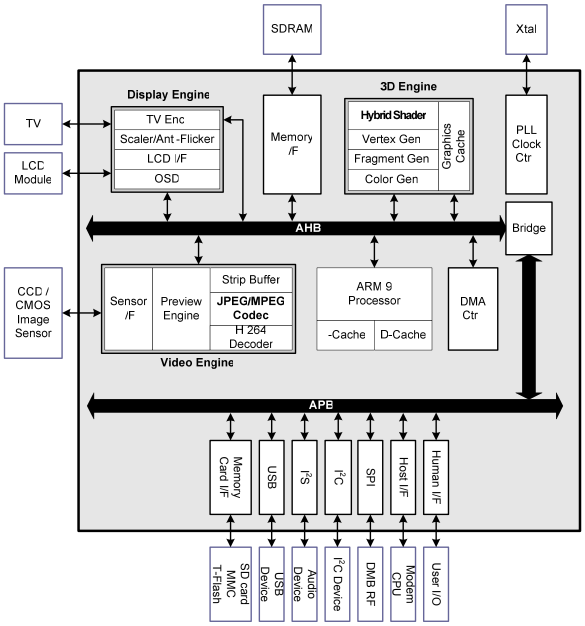
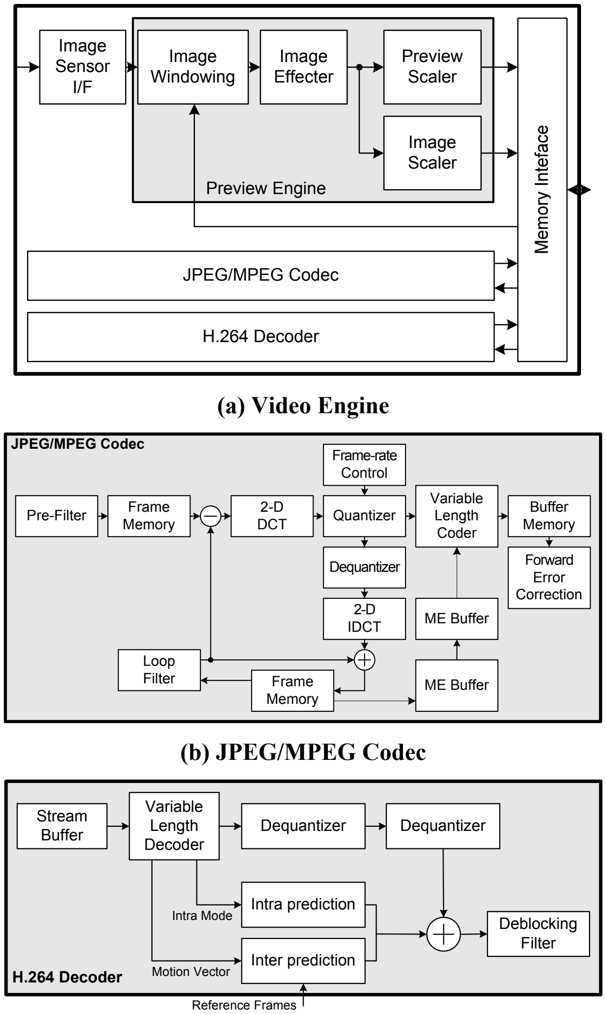


**Fig. 1. SoC System Block Diagram**

The SoC is attached as a multimedia application processor in a mobile system. The system processor such as a baseband processor can access the SoC through host interface using 16bit asynchronous SRAM protocol. The SoC uses an external SDRAM as a data memory. The graphics data such as depth data and texture images are stored in the SDRAM. And the frame buffer of 3-D graphics engine and video engine are also allocated in the SDRAM. To reduce physical footprint in mobile system, the SDRAM is stacked in the package.

The video engine is employed to support multimedia application such as DMB or digital camera [Fig.2.]. It performs MPEG4 codec, H.264 decoding and JPEG codec. Both of the MPEG4 codec and JPEG codec use DCT, IDCT, VLC and VLD units and these blocks are shared by two codecs to reduce silicon area and power consumption. According to the applications, the merged codec hardware is configured to JPEG codec or MPEG4 codec and the unused hardware block is clock gated to reduce the power consumption. The H.264 decoder consists of a contents-addressable-variable-length-decoder, an inverse-transform, a motion compensator and de-blocking filter. The video engine supports JPEG codec with up to 3Mpixels image sensor, MPEG4 codec at simple profile Lv.0~3 of 30fps for CIF and H.264 decoding at baseline Lv.0~3 of 30fps for CIF. During full video operation, it consumes less than 152mW at 1.2V supply voltage and 48MHz operating frequency.



**(a) Video Engine**

**(b) JPEG/MPEG Codec**

**(c) H.264 Decoder**

**Fig. 2. Video Engine**

# 3. PROGRAMMABLE 3D GRAPHICS ENGINE

## 3.1 3-D Graphics Engine Architecture

The 3D graphics engine (3DGE) consists of Unified Shader (USH), Vertex Generator (VG), Fragment Generator (FG), Pixel Generator (PG), Matrix/Quaternion-Vector Generator (MG) and graphics caches as shown in Fig. 3.

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 shader and the pixel shader are merged into the single USH. With the USH in 3DGE, the silicon area and power consumption are reduced by 35% and 28%, respectively. The dedicated texture engine performs texture address generation, texture fetch and filtering. The vertex generator, fragment generator and pixel generator perform clipping, shading and blending, respectively. To reduce the power

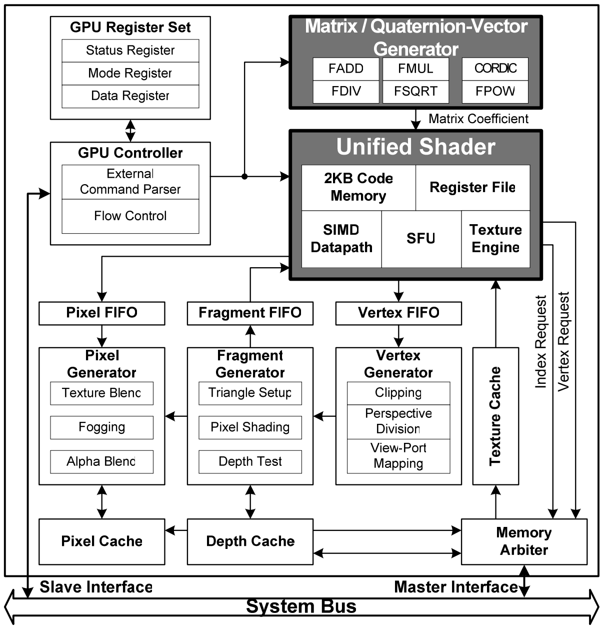


**Fig. 3. Programmable 3D Graphics Engine**

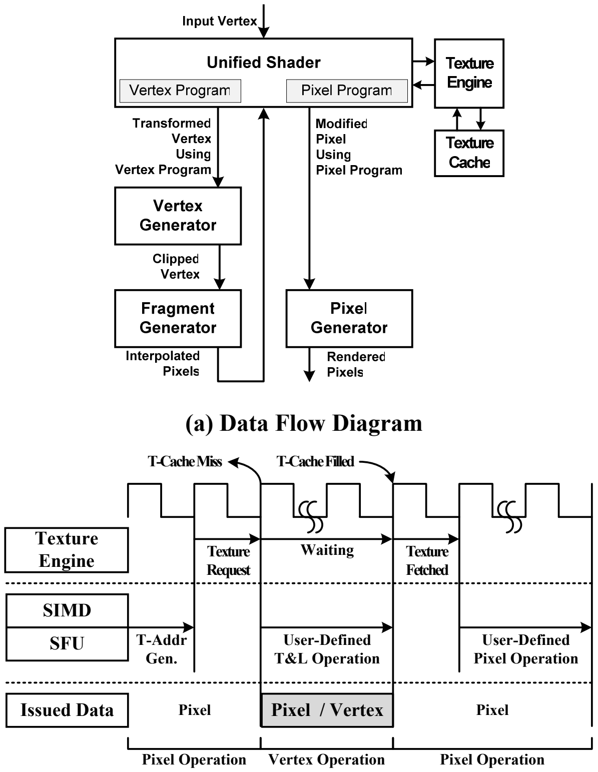
consumption during rendering operation, the pixel-level clock-gating [4] is employed.

To enhance the graphics performance, the MG is employed. Since the floating-point matrices, which generation consume thousands of cycles on RISC [4], are used by USH, the speed of matrix generation restricts the total 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.

## 3.2 Pixel-Vertex Multi-Threading

Fig 4.-(a) shows a data flow diagram of programmable 3D graphics operation. In the programmable-pipeline mode, the USH processes both vertex program and pixel program. 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, in the case of texture cache miss, the USH wastes tens of cycles without any operation until texture cache to be filled and it degrades graphics performance. Since the texture engine is independent of SIMD datapath and SFU, the SIMD datapath and SFU can calculate other vertex while texture engine is halting. To enhance graphics performance, the USH adopts a Pixel-Vertex-Multi-Threading (PVMT), which enables to utilize datapaths in parallel. When the texture cache miss is occurred during programmable pixel operation, the USH changes its operation mode to programmable vertex operation and a new vertex is issued to USH. After finishing the programmable vertex operation, the USH move back to the programmable pixel operation and finish the pixel operation. [Fig 4.-(b)] By adopting PVMT, the datapaths of USH are fully utilized and it improves graphics performance further.



**(a) Data Flow Diagram**

**(b) Pixel-Vertex Multithreading**

**Fig. 4. Pixel-Vertex Multi-Threading**

# 4. UNIFIED SHADER

## 4.1 Internal Architecture

The USH 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 Fig. 5.

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) [6].

For streaming graphics processing, USH contains multiple register files—input registers (IR), output vertex registers

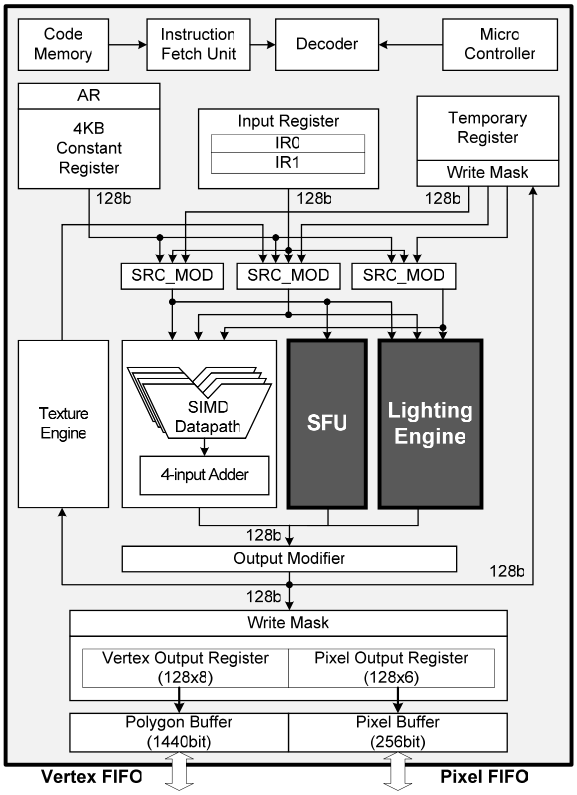


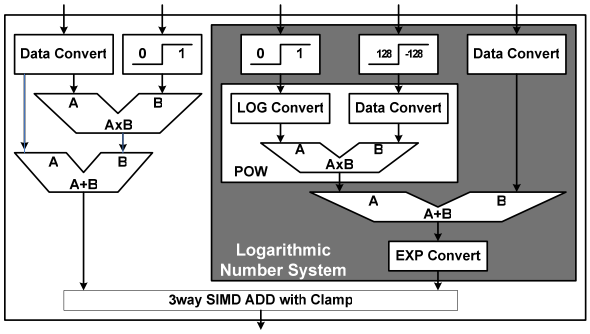
**Fig. 5. Unified Shader Architecture**

(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. 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.

## 4.2 Lighting Engine (LE)

Since the lighting calculation is the most complex calculation during vertex operation, the LE is employed to improve vertex throughput with low power consumption. The LE of Fig. 6-(a) has the LNS datapath for the power (POW) operation of the specular light and ordinary datapath for the ambient and the diffuse light calculation.

The specialized TLT instruction is proposed to enhance lighting calculation. It is very useful for lighting calculations. 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. Fig. 6-(b) presents the comparison of the lighting instruction sequence 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 provides 9.1Mvertecies/sec, 2.1 times higher performance compared with previous implementation [5].



**(a) Lighting Engine**

| | |
|---|---|
| C[1].x | = Pre-multiplied diffuse light color & diffuse material |
| C[1].y | = Pre-multiplied ambient light color & diffuse material |
| C[2] | = Specular Color |
| C[3].x | = Specular Power |
| N' | = Transformed Normal Vector |
| H | = Half-Angle Normal Vector |

**Lighting Equation**

$$O[COL].xyz = \underbrace{C[1].y}_{\text{Ambient}} + \underbrace{(N' \bullet H) \times C[1].x}_{\text{Diffuse}} + \underbrace{(N' \bullet H)^{C[3].x} \times C[2]}_{\text{Specular}}$$

**Instruction Sequence**

| | | |
|---|---|---|
| DP3 | R1.x, Ldir, N'; | R1.x = Ldir • N' |
| DP3 | R1.y, H, N'; | R1.y = H • N' |
| MOV | R1.w, C[3], x; | |
| LIT | R2, R1; | |
| MAD | R3, C[1].x, R2.y, C[1].y; | R2.y = (N'•H) |
| MAD | O[COL].xyz, C[2], R2.z, R3; | R2.z = $(N'•H)^{C[3]}$ |

**Previous Implementation**

| | |
|---|---|
| DP3 | R1.x, Ldir, N'; |
| DP3 | R1.y, H, N'; |
| MOV | R1.w, C[3], x; |
| MOV | R2, C[2]; |
| **TLT** | **O[COL].xyz, R1, C[1], R2;** |

**Proposed Lighting Engine**

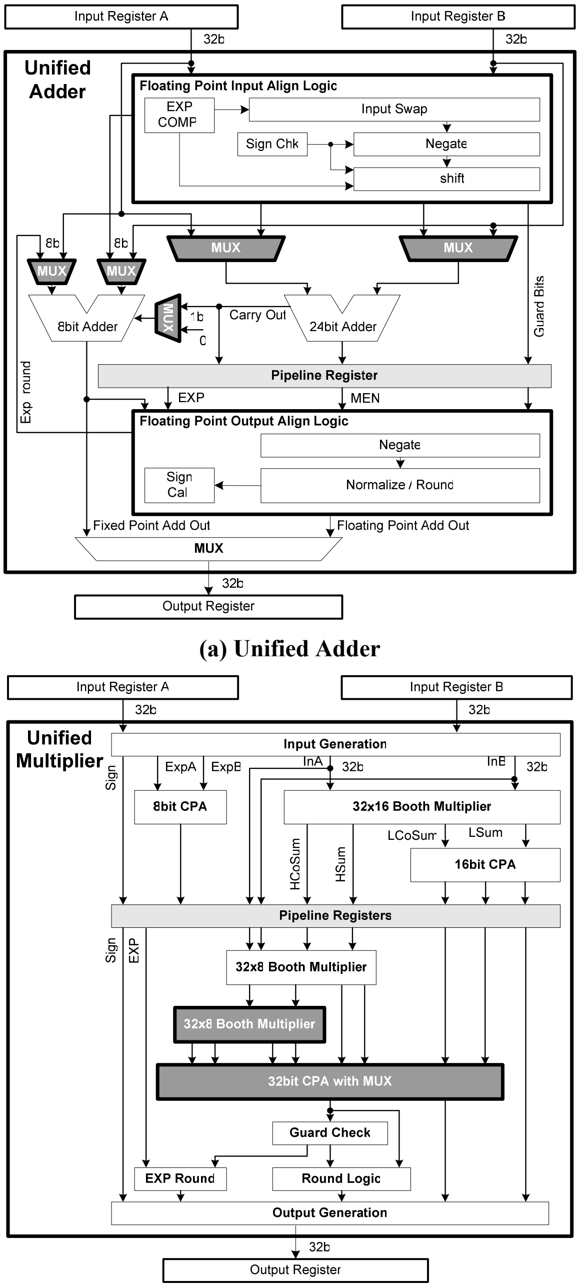
**(b) Instruction Sequence Comparison**

**Fig. 6. Lighting Engine**

## 4.3 Unified Datapath

In mobile applications, fixed-point data is enough for rendering operation, and floating-point is used only for geometry operation. Because the USH performs both of the geometry operation and rendering operation in the single hardware, the SIMD datapath and SFU are required to handle both floating-point and fixed-point data.

The unified adder (UA) splits 32bit adder into 24bit adder and 8bit adder. And by adding MUXs to select operands of adders as shown in Fig. 7-(a), both floating-point add and fixed-point add are calculated by the single 32bit adder.



(a) Unified Adder

(b) Unified Multiplier

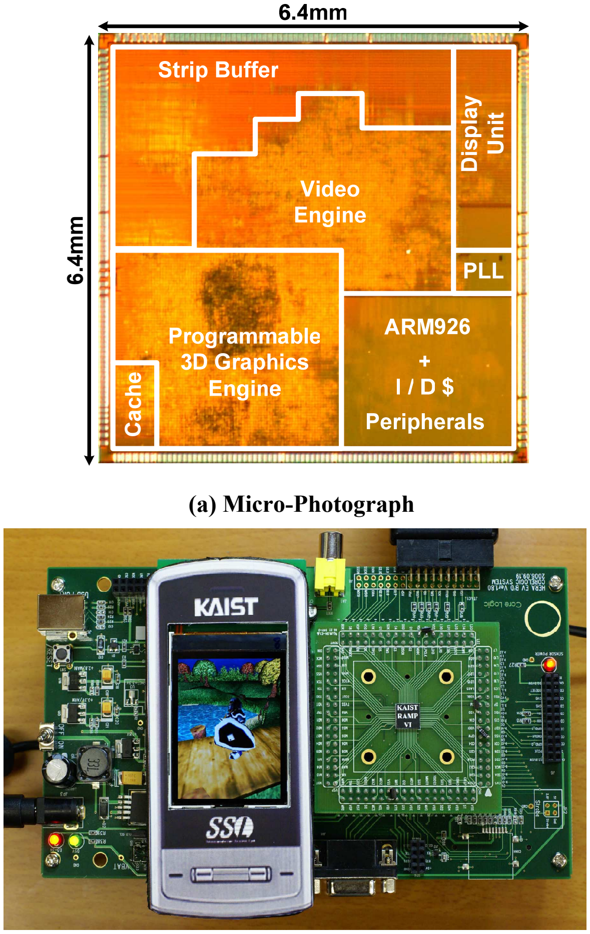
Fig. 7. Unified Datapath

The UA calculates floating-point addition with 2cycle latency and 1cycle throughput, and fixed-point with 1cycle latency and 1cycle throughput. The floating point align logics prevent data transition during fixed-point addition to reduce dynamic power consumption in UA. The unified multiplier (UM) consists of common 24bit multiplier and optional 8bit multiplier as shown in Fig. 7-(b). The final 32x8 multiplier is conditionally enabled and the CPA chain selects the input between 24bit result and 32bit result. The UM calculates both floating-point MUL and floating-point MUL with 2cycle latency and 1cycle throughput.

# 5. CHIP IMPLEMENTATION

## 5.1 Implementation Results

The SoC is fabricated by a 0.13μm 7-metal CMOS logic process. It contains 18.6M transistors including 128KB SRAM in 6.4x6.4mm$^2$. It provides MPEG4 codec and H.264 decoding, JPEG codec and fully programmable 3D graphics. During 3D graphics application, the USH, logarithmic datapaths and low power schemes reduce the power consumption and the SoC consumes less than 195mW at 100MHz operating frequency and 1.2V supply voltage. And the SoC consumes less than 152mW for video operation at 1.2V and 48MHz. The Table 1 summarizes chip features and Fig.8 shows the chip micro-photograph and the evaluation system. Fig.9 shows the 3D graphics performance comparison with that of the previous implementations [3-5]. The SoC provides the fully programmable 3D graphics and it improves 2.1 times vertex fill rate compared with previous implementation and



(a) Micro-Photograph

(b) Evaluation System Board

Fig. 8. Implementation Results

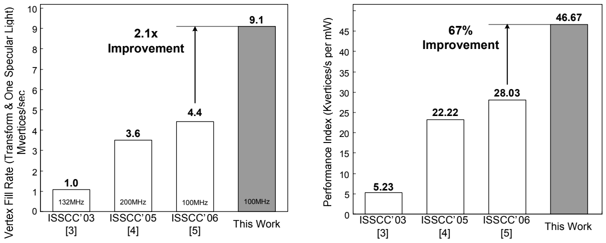
**Table 1. Chip Feature Summary**

| | | |
|---|---|---|
| Process Technology | | 0.13µm 7M CMOS |
| Power Supply | | 1.2V |
| Transistor Counts | | 18.6M Transistors ( including 128KB SRAM ) |
| Die Size | | 6.4mm x 6.4mm |
| Operating Frequency | | 100MHz |
| Power Consumption | | 152mW @ Video Processing<br>195mW @ Full 3D Graphics Processing |
| Package | | 144pin CABGA |
| Performance | Image Processing | Up to 3M pixels Camera Resolution Support<br>Real-Time Baseline JPEG Encoding / Decoding |
| | Video Processing | MPEG : CIF 30 fps Codec @ SP Lv.0~3<br>H.264 : CIF 30 fps Decode @ BL Lv.0~3 |
| | Geometry | 9.1Mvertices/s (Including Full OpenGL Lighting) |
| | Rendering | 100Mpixels/s, 400Mtexels/s |
| Graphics Functions | Programmability | Programmable Vertex Shader<br>Programmable Pixel Shader |
| | Standard API | OpenGL\|ES ver. 1.1<br>OpenGL\|ES ver. 2.0 |
| | Screen Resolution | Up to 1024 x 1024 pixels |

67% in graphics performance normalized by power consumption with the help of TLT instruction and the logarithmic LE.

# 6. CONCLUSIONS

A low power multimedia SoC is designed and implemented for mobile devices. It integrates MPEG4/JPEG codec, H.264 decoder and fully programmable 3D graphics engine. The USH provides a fully programmable 3D graphics with 35% area reduction and 28% power reduction. Logarithmic lighting engine and specialized lighting instruction achieves 9.1MVertices/s vertex fill rate and PVMT improves graphics performance further. The SoC consumes less than 195mW at 1.2V supply voltage and 100MHz operating frequency for 3D graphics and less than 152mW at 1.2V supply voltage and 48MHz operating frequency for video operation. The unified shader and merged JPEG/MPEG 4 codec reduces the silicon area and the SoC consumes 6.4mm x 6.4mm in 0.13 µm CMOS logic process.



| | [3] | [4] | [5] | **This Work** |
|---|---|---|---|---|
| Operation Frequency (MHz) | 132/33 | 200 | 100 | **100** |
| Vertex Fill Rate (Mvertices/s) | 1.1 | 3.6 | 4.4 | **9.1** |
| Power Consumption (mW) | 210 | 155 | 157 | **195** |
| Programmability | N/A | Geometry Only | Geometry Only | **Geometry Rendering** |

**Fig. 9. Performance Comparison**

# 7. REFERENCES

[1] J.H. Sohn, et. al., "Low Power 3D Graphics Processors for Mobile Terminals", IEEE Communication Magazine, Dec. 2005

[2] T.M. Liu, et al., "A 125□W, Fully Scalable MPEG-2 and H.264/AVC Video Decoder for Mobile Applications", ISSCC Digest of Technology papers, pp. 402-403, 2006

[3] R. Woo, et al., "A 210mW Graphics LSI Implementing Full 3D Pipeline with 264Mtexels/s Texturing for Mobile Multimedia Applications", ISSCC Digest of Technology Papers, pp. 44-45, 2003

[4] J.H Sohn, et al., "A 155-mW 50-Mvertices/s Graphics Processor With Fixed-Point Programmable Vertex Shader for Mobile Applications", IEEE Journal of Solid State Circuits, Vol 41, No. 5, pp.1081-1091, May, 2006

[5] C.H. Yu, "A 120Mvertices/s Multi-threaded VLIW Vertex Processor for Mobile Multimedia Applications", ISSCC Digest of Technology Papers, pp.408-409, 2006

[6] H. Kim, et al., "A 231MHz, 2.18mW 32bit Logarithmic Arithmetic Unit for Fixed-Point 3D Graphics System", IEEE Journal of Solid-State Circuits, Vol 41, NO. 11, pp2373-2381, November, 2006

[7] M. Doggett, "Xenos: XBOX360 GPU", ACM Eurographics2005