# A Fixed-point Multimedia Co-processor with 50Mvertices/s Programmable SIMD Vertex Shader for Mobile Applications

Ju-Ho Sohn[(1)], Jeong-Ho Woo[(1)], Ramchan Woo[(2)] and Hoi-Jun Yoo[(1)]

(1) Semiconductor System Laboratory, Department of EECS
Korea Advanced Institute of Science and Technology, Daejeon, Korea.
(2) Imaging and Audio Group, Texas Instruments Incorporated, Dallas, Texas, USA
(sohnjuho@eeinfo.kaist.ac.kr)

## Abstract:

A fixed-point multimedia co-processor is designed and integrated into an ARM-10 based mobile graphics processor for portable 2-D and 3-D multimedia applications. The user-programmable SIMD vertex shader with ARM-10 co-processor architecture realizes advanced 3-D graphics algorithms and various multimedia functions. Different from conventional ARM co-processor architecture, the multimedia co-processor implements dual operations, by which parallel and streaming multimedia processing is enabled in mobile applications. For low power consumption, fixed-point SIMD Datapath is designed with instruction-wise clock gating. The co-processor takes 10.2mm$^2$ in 0.18μm 6-metal standard CMOS logic process and achieves 50Mvertices/s graphics performance with 75.4mW power consumption.

## 1. Introduction

As the 3-G handheld devices such as PDAs and smart cell-phones are getting more popular, more multimedia functions are required in wireless terminals. And their applications are already moving to various 2-D multimedia functions and even to the realtime 3-D graphics. Recently, several researches on multimedia architectures have tried to increase the mobile graphics capabilities in mobile applications. In the previous works, they did not integrate all required hardware blocks like the vertex shader [1] and showed lack of processing parallelism for streaming graphics data [2]. They provided the fixed functionalities and were revealed to show low performance. Also, the conventional floating-point architectures intentionally decreased the performance by lowering the operating frequency to meet the limited power budget [3][4].

Since users watch the 3-D graphics images on a small screen of mobile devices very close to their eyes, every pixel in mobile applications should be drawn with higher quality by advanced graphics algorithms than that in a PC system [5]. These advanced graphics algorithms require the programmability such as DirectX and OpenGL shading language extensions in graphics hardware. Moreover the programmability can allow the single hardware to provide various multimedia functions beyond 3-D graphics such as MPEG-4 video.

In this work, we designed and implemented a fixed-point multimedia co-processor with programmable SIMD vertex shader for the ARM-10 based mobile graphics processor [6]. The fixed-point datapath is used to reduce the power consumption because only simple integer arithmetic circuits can be used in complex multimedia operations. We optimize the circuits and architecture so that ARM-10 co-processor architecture with user-programmable vertex shader can realize advanced 3-D graphics algorithms and various multimedia functions while satisfying the requirements of the battery lifetime and the system resources in mobile devices.

## 2. System Architecture

The system architecture of the proposed multimedia co-processor is shown in Fig. 1. The multimedia co-processor is a 128-bit 4-way SIMD co-processor of the ARM-10 processor architecture [7]. It is connected to ARM-10 main processor through the co-processor interface and can perform general integer and fixed-point SIMD arithmetic operations and 3-D graphics functions such as geometry transformation and lighting calculation. The co-processor architecture reduces system bus bandwidth by avoiding complex bus arbitrations. It also enables easy integration with efficient programmability.

The co-processor consists of two parts – control and datapath. The control part contains a co-processor interface unit and an instruction control unit. It generates the control signals of the datapath elements by decoding

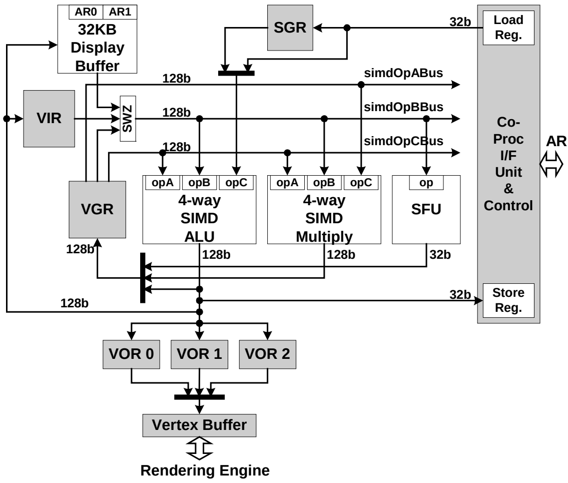


**Figure 1: Multimedia Co-processor Architecture**

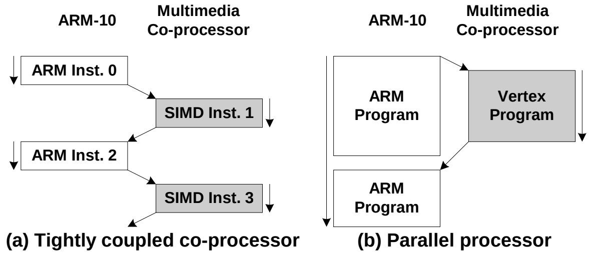
the current instruction. In the datapath part, there are 4-way SIMD arithmetic and logic unit (ALU) and SIMD multiplier unit that are responsible for all SIMD arithmetic operations such as addition and multiplication. Special function unit (SFU) is responsible for reciprocal (RCP) and reciprocal square root (RSQ) operations. Most of operations are performed in 32-bit fixed-point numbers, and achieve the single cycle throughput. Display buffer, implemented as 32kB SRAM stores frequently accessed multimedia data and graphics constants. To enable streaming multimedia processing, the co-processor contains multiple register files – input vertex registers (VIR), output vertex registers (VOR) and general SIMD registers (VGR). The co-processor also has internal vertex buffer that can interface with hardware rendering engine for additional graphics rendering operations [8].

## 3. Dual Operations

Different from conventional ARM co-processor architecture [9], the proposed multimedia co-processor has dual operating states as shown in Fig. 2.

(1) Tightly coupled co-processor (TCC): In this state, the multimedia co-processor is a normal ARM-10 co-processor. The instructions of the co-processor are issued in the instruction stream of the main processor as extended co-processor instructions, and they are executed in lock step with pipeline of the main processor. TCC state implements general integer and fixed-point SIMD data processing instructions and all instructions can be executed conditionally like other ARM instructions.

(2) Parallel Processor (PP) [10]: In this state, the co-processor is an independent processor and can operate without control of ARM-10 processor. PP state has a separate graphics instruction set from general SIMD instructions of TCC state. The co-processor executes the independent vertex program codes while ARM-10 processor performs main application program or even enters into cache miss state. Various user-defined vertex processing such as geometry transformation and lighting calculation can be performed for current vertex input during next vertex fetch of ARM-10 processor.



(a) Tightly coupled co-processor (b) Parallel processor

**Figure 2: Dual Operations**

Fig. 3 shows the block diagram of control part for dual operations. Vertex program control unit (VPCTRL) issues the graphics instructions without control of ARM-10 processor from 2kB code memory. The general SIMD instructions are transferred through the co-processor interface. The control register determines the operating state. The two operating states – TCC state and PP state share all hardware blocks except instruction fetch units. To maintain the communication protocol of ARM-10 co-processor interface, the multimedia co-processor drives the co-processor busy signal (CPbusy) in PP state so that next co-processor instruction from the main processor stands by for synchronization.

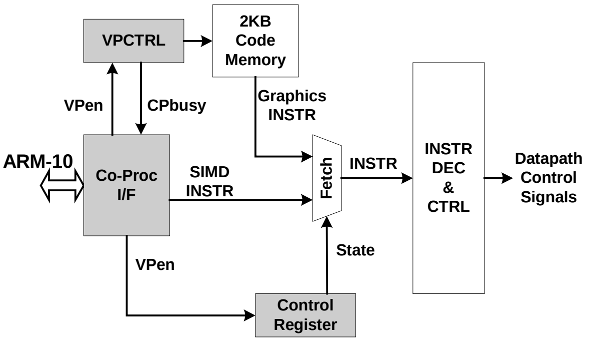


**Figure 3: Block Diagram of Control Part**

## 4. Co-processor Pipeline

The eight-stage single-issue pipeline of the co-processor is illustrated in Fig. 4. The fetch stage transfers one of the general SIMD instructions and the graphics instructions from the co-processor interface and the code memory, respectively to the control unit. For programmable shading, operands of the SRAM display buffer and SIMD register files are accessed at the same time in decode stage. The SRAM address is generated in the early stage of the pipeline, the issue stage. In execution stage, there are three separated pipelines – SIMD ALU pipeline, SIMD multiply pipeline and SFU pipeline. To reduce the design complexity, register-forwarding logic between pipeline stages is used only in VGR.

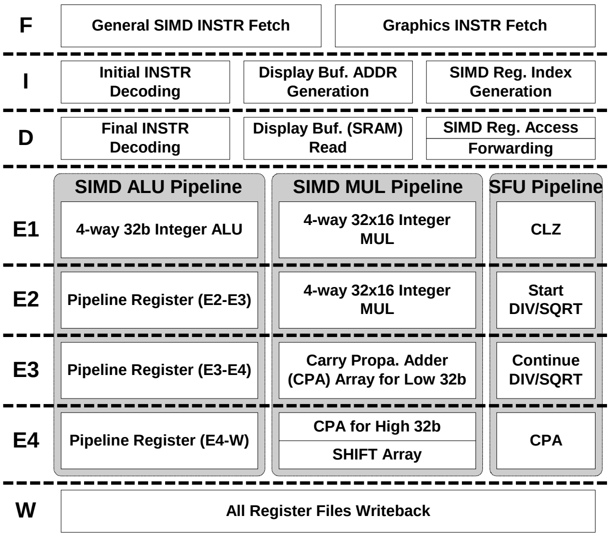


**Figure 4: Co-processor Pipeline Structure**

In order to maximally save the power consumption, the clock gating is performed as instruction-by-instruction basis as shown in Fig. 5. By the definition of the ARM-10 co-processor interface, ARM-10 processor must drive co-processor instruction valid (CPINSTV) signal to the co-processor only when the current instruction issued from ARM-10 processor is the valid co-processor instruction. Using CPINSTV, the clock signals of SIMD register files can be gated off when the write operations of the register files are not required. CPINSTV also reduces power dissipated in the datapath of SIMD arithmetic units by eliminating the unnecessary signal transitions.

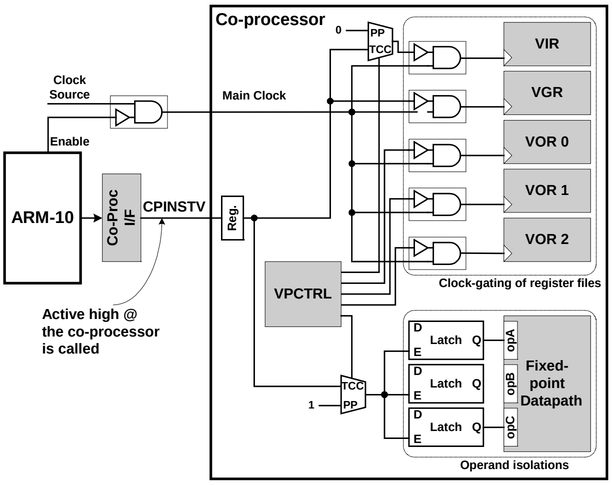


**Figure 5: Instruction-wise Clock Gating**

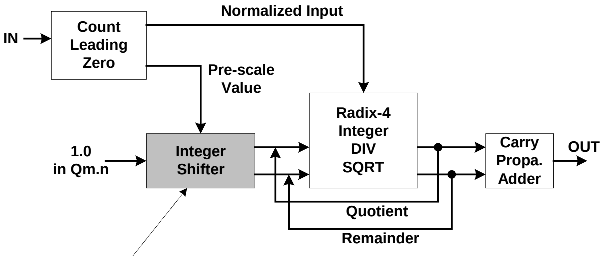
## 5. Fixed-point Datapath

Most of 2-D and 3-D multimedia applications require real number representation to support various algorithms. In the proposed multimedia co-processor, fixed-point number representation is used instead of floating-point number for low power consumption. All datapath elements are designed to perform fixed-point arithmetic operations efficiently by using only simple integer arithmetic circuits.

Fig. 6 shows the block diagram of the 32-bit fixed-point multiplier in the SIMD multiply unit. Two stage 32-by-16 integer multipliers with integer shifters for fixed-point conversion achieve the single cycle throughput for fixed-point multiply and accumulate (MAC) operation. In addition, fast 4-cycle matrix transformation (TRFM) is implemented in the SIMD multiply unit. By broadcasting vector elements of input vertex, TRFM can be calculated by the first MUL and the following three MAC operations. However, fixed-point MUL and MAC operations require two cycle integer multiplications and two cycle integer additions, leading latency to be 4-cycle. To resolve data dependency between these MUL and MAC operations, it is allowed that the intermediate value of the integer multipliers can be bypassed to accumulate input of the integer adder. By this scheme, the proposed co-processor shows 50Mvertices/s peak graphics performance for parallel projection at 200MHz operating frequency.

SFU (Fig. 7) calculates the square root and division by using 32-bit radix-4 combined integer division and square root unit. It calculates fixed-point result from fixed-point input number. Integer shifter in SFU pre-scales the input fixed-point number to intermediate 64-bit integer format before actual division and square root operations. Since output fixed-point number is 32-bit value, only MSB 32-bit of the intermediate 64-bit integer value is calculated after counting-leading-zero (CLZ) operation.

*For Qm.n Fixed-point Format*



- Scale up fixed-point dividend to 64b space

| OP | Pre-scale Value |
|---|---|
| RCP | # of Leading Zero |
| RSQ | n/2 + # of Leading Zero |

**Figure 7: Special Function Unit**

Fig. 8 shows the block diagram of the SIMD ALU in the fixed-point datapath. It can calculate all of the arithmetic and logic operations including byte shuffle, data packing and operand alignment. Since 32-bit fixed-point number

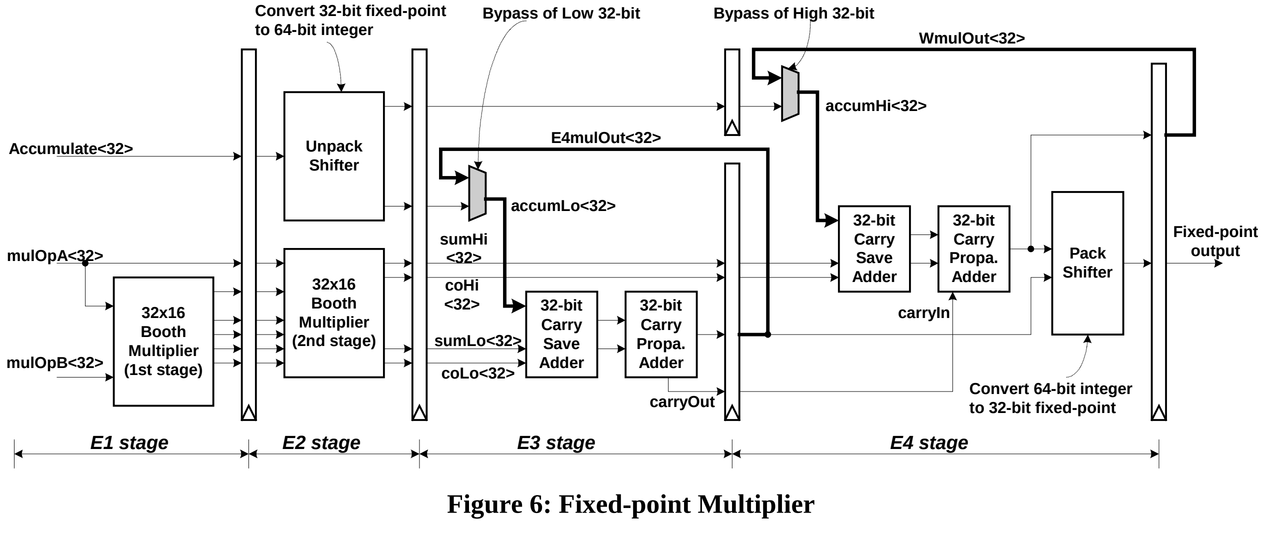
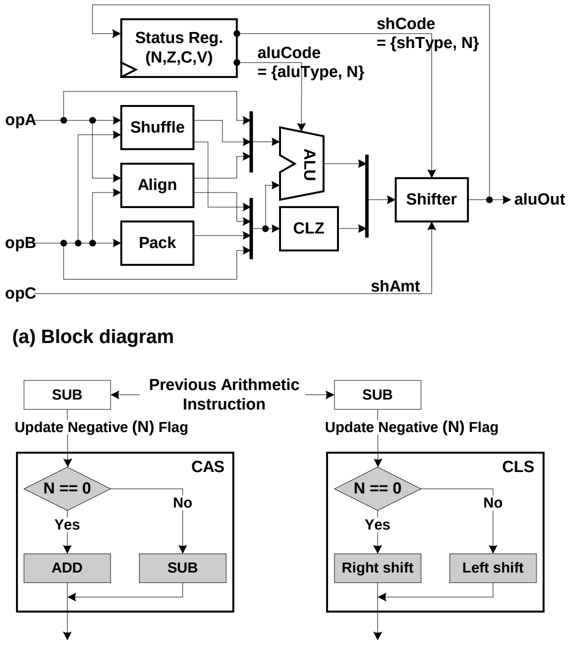


**Figure 6: Fixed-point Multiplier**

is represented in a typical 32-bit integer type, integer adder and shifter circuits are used for calculation of the fixed-point numbers.

Although fixed-point arithmetic provides robust performance in mobile multimedia processing, various multimedia applications such as physical calculations still require enhancement of dynamic range in real number representation. In the co-processor, two special instructions – controlled ADD/SUB (CAS) and controlled logical shift (CLS) are added for efficient software floating-point emulation as shown in Fig. 8.(b). In order to enhance SIMD parallelism in software programming of floating-point routines, the CAS and the CLS instructions change the control flow instructions to single cycle SIMD arithmetic operations. With floating-point emulation, the proposed co-processor shows 80Mflops peak floating-point performance at 200MHz operating frequency.



(a) Block diagram

(b) Two instructions (CAS, CLS) for floating-point emulations

Figure 8: SIMD ALU

## 6. Conclusion

The proposed multimedia co-processor is fabricated by 0.18μm 6-metal standard CMOS logic process and integrated into the ARM-10 based mobile graphics processor. As described in Table 1, the co-processor takes 10.2 mm$^2$ and consumes 75.4mW in the continuous calculations of full 3-D geometry operations. It utilizes ARM-10 co-processor architecture with dual operations for advanced 3-D graphics functions and various multimedia operations, while achieving the low power consumption by fixed-point datapath and instruction-wise clock-gating. The peak graphics performance is 50Mvertiecs/s at 200MHz operating frequency. Fig. 9 shows the die photograph and system evaluation board. The evaluation board equipped with implemented chip demonstrates realtime 3-D graphics images with various graphics algorithms.

| Process | 0.18 um 6-Metal CMOS |
|---|---|
| Power supply | 1.8V |
| Operating frequency | 200MHz |
| Die size | 10.2mm$^2$ |
| Processor Performance | 1000MIPS (Integer) |
|  | 80MFLOPS (Floating-point) |
| Graphics Performance | 50Mvertices/s<br>(Geometry transformation) |
|  | 3.6Mvertices/s (sustaining)<br>(Full 3-D geometry operations) |
| Power Consumption | 75.4mW (With lighting) |
|  | 60.3mW (Transform only) |

Table 1: Co-processor Features

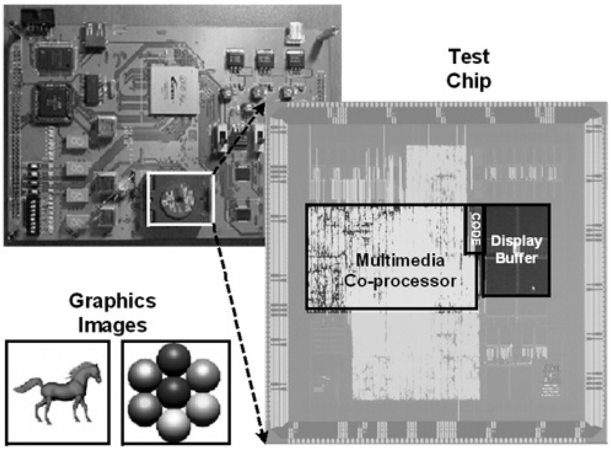


Figure 9: Die Photograph and Evaluation Board

## References:

[1] Chi-Weon Yoon, et al, "An 80/20MHz 160mW Multimedia Processor Integrated With Embedded DRAM, MPEG-4 Accelerator, and 3D Rendering Engine for Mobile Applications," ISSCC, pp.142-143, 2001

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

[3] Masatoshi Imai, et al, "A 109.5mW 1.2V 600Mtexels/s 3-D graphics engine," ISSCC, pp. 332-333, 2004

[4] Masatoshi Kameyama, et al, "3-D LSI core for mobile phones – Z3D," Graphics Hardware, pp.60-67, 2003

[5] Tomas Akenine-Moller, et al, "Graphics for the masses: A hardware rasterization architecture for mobile phones," SIGGRAPH, pp.801-808, 2003

[6] Ju-Ho Sohn, et al, "A 50Mverties/s graphics processor with fixed-point programmable vertex shader for mobile applications," ISSCC, pp. 192-193, 2005

[7] Ju-Ho Sohn, et al, "A Programmable Vertex Shader with Fixed-point SIMD Datapath for Low Power Wireless Applications," Graphics Hardware, pp.107-114, 2004

[8] Ramchan Woo, et al, "A Low Power 3D Rendering Engine with Two Texture Units and 29Mb Embedded DRAM for 3D Multimedia Terminals," ESSCIRC, pp.53-56, 2003

[9] Intel Wireless MMX Technology, available: http://www.intel.com

[10] Prashant P. Gandhi, "SA-1500: A 300MHz RISC CPU with Attached Media Processor," HotChips 10, 1998