

A 120mW Embedded 3D Graphics Rendering Engine with 6Mb Logically Local Frame-Buffer and 3.2GByte/s Run-time Reconfigurable Bus for PDA-Chip

Ramchan Woo, Chi-Weon Yoon, Jeonghoon Kook, Se-Joong Lee,
Kangmin Lee, Yong-Ha Park and Hoi-Jun Yoo

Semiconductor System Laboratory, Department of Electrical Engineering,
Korea Advanced Institute of Science and Technology, Taejon, Korea

Abstract

An embedded 3D graphics rendering engine (E3GRE) is implemented as a part of a mobile PDA-chip. 6Mb embedded DRAM (eDRAM) macros attached to 8-pixel-parallel rendering logic are logically localized with 3.2GByte/s run-time reconfigurable bus, by which the area is reduced by 25%. Polygon-dependent access to eDRAM macros with line-block mapping reduces the power consumption by 70% with the read-modify-write data transaction. E3GRE with 2.22Mpolygons/s drawing speed was fabricated using 0.18 μ m CMOS embedded memory logic technology. Its area and power consumption are 24mm² and 120mW, respectively.

1. Introduction

As the mobile electronics market grows, hand-held devices such as palm-sized PC or Personal Digital Assistance (PDA) are becoming popular and more processing power of these devices is getting required because the market is moving from text-based Personal Information Management (PIM) to multimedia applications such as IMT-2000 terminals. Recently, a multimedia PDA-chip, which contains a 32bit RISC core, a MPEG-4 video decoder, and a 3D graphics rendering engine with embedded-DRAM (eDRAM), was presented [1]. In order to draw 3D primitives on an LCD screen in the PDA-chip, the embedded 3D graphics rendering engine (E3GRE) must be small in size and low in power consumption while sustaining the high polygon drawing rate. Local frame-buffer architectures [2, 3, 4] shown in fig. 1(a), in which each local memory is tightly coupled with corresponding pixel processor (PP), provides high performance because required bandwidth for parallel calculation of pixel data is obtained by local-route. Besides, the power consumption is low because only necessary memories can be selectively activated. In these architectures, however, poor cell-efficiency of eDRAM increases the chip area. Although a global frame-buffer architecture [5] shown in fig. 1(b), in which required pixel data are obtained from wide-bus of single eDRAM by bank-interleaving, is a good candidate in terms of cell-efficiency, it wastes the power because unnecessary data are transferred together with required ones through long on-chip wires. In this paper, we propose a logically local frame-buffer architecture with eDRAM macros and run-time reconfigurable bus to achieve

low power, small area, and high polygon drawing rate.

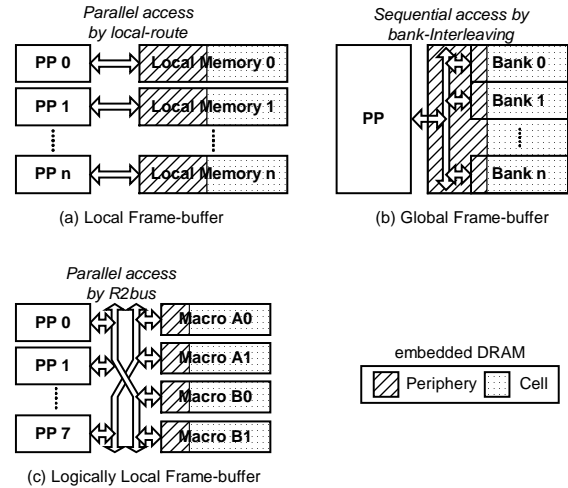


Fig. 1 : Architecture Diagram of Embedded Frame-Buffer

2. Logically Local Frame-Buffer

Fig. 2 shows the block diagram of the E3GRE. One edge processor (EP) calculates pixel data along edges of 8x8-clipped polygon and broadcasts them to 8 pixel processors (PP) which fill horizontal pixels inside polygon in parallel. 6Mb eDRAM macros, which form a ‘physically global’ frame-buffer, are attached to the rendering logic through the run-time reconfigurable bus (R2bus). This global frame-buffer reduces the area by enhancing the cell-efficiency of eDRAM. The R2bus provides ‘logically local’ frame-buffers [fig. 1(c)] to 8 PPs by changing both inter and intra bus between eDRAM macros and rendering logic at every cycle to get high drawing speed with low power consumption.

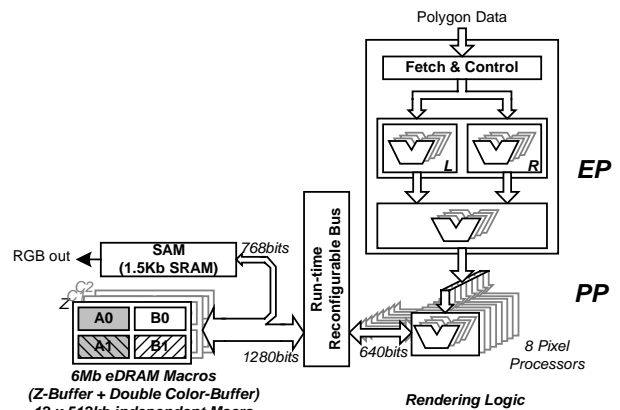


Fig. 2 : Block Diagram of Embedded 3D Graphics Rendering Engine

3. Line-Block Memory Mapping

Fig. 3 shows the proposed line-block mapping. Each 8x1 screen pixels compose a line-block and adjacent line-blocks are mapped into sub-wordlines of different eDRAM macro units. Therefore, 4 macro units (A0, B0, A1, B1), each of which contains one eDRAM macro for depth-buffer and two eDRAM macros for double color-buffer, are necessary to cover the screen area, and each line-block provides 8 pixel-data (320bits) in parallel taking advantage of the wide-bus of eDRAM.

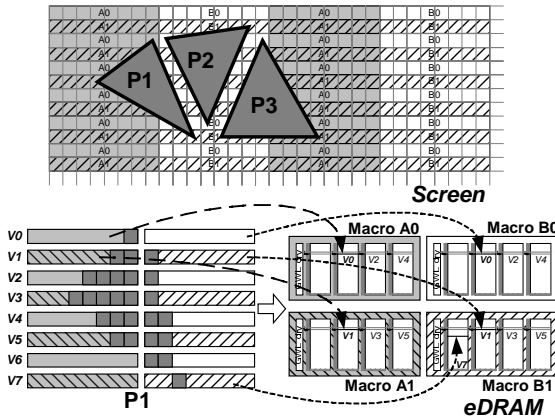


Fig. 3 : Line-Block Memory Mapping

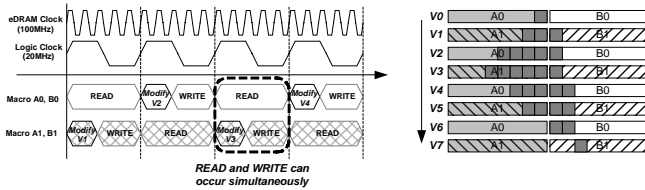


Fig. 4 : Simultaneous and Continuous Read-Modify-Write

Alternative memory mapping (A0,B0 and A1,B1) in the vertical direction enables simultaneous and continuous read-modify-write operation as described in fig. 4. At the first logic cycle, the data for line V2 in fig. 4 are read from the upper macro units (A0, B0). Then these data are subsequently modified in the rendering logic and written back to the same upper macro units at the second logic cycle. At the same second cycle, the data for V3 are read from the lower macro units (A1, B1). Therefore, rendering logic continuously modifies the data by simultaneous memory READ and WRITE operations.

4. Polygon-Dependent eDRAM Access

The power is wasted in a conventional global frame-buffer [fig. 5] due to unnecessary data transaction. Therefore, we use three low-power eDRAM access methods with line-block activation to reduce the power consumption; Selective Macro Activation (SMA), Partial Wordline Activation (PWA), and Partial I/O Activation (PIA). As shown in fig. 3, 8x8-clipped polygon falls into only left macro units (A0, A1), only right macro units (B0, B1) [P2 in fig. 3], or both of them [P1, P3]. Moreover, as for a polygon which falls into both of them

such as P1, all vertical-lines do not require both macro units. In fig. 6, a top line requires only left macro unit (A0) and two bottom lines require only right macro unit (B0, B1). Therefore, if only required macro units are selectively activated by SMA, the power can be reduced. Simulation result shows that more than 90% lines in polygons require only one bank macro unit. And, only necessary sub-wordline for line-block instead of full wordline is activated by PWA to reduce the power of eDRAM core operation as shown in fig. 7. Lastly, PIA is used [fig. 8] to eliminate unnecessary I/O bus transaction which consumes large power because of long capacitive wires between eDRAM and logic.

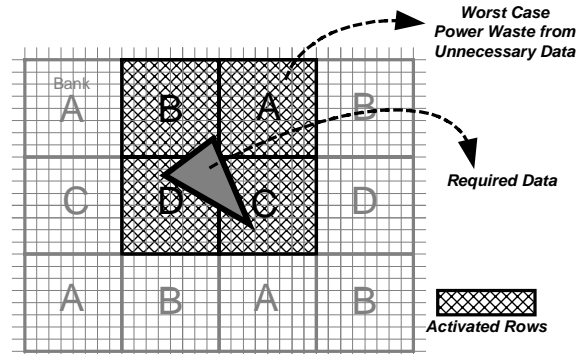


Fig. 5 : Power Waste in Conventional Global Frame-Buffer

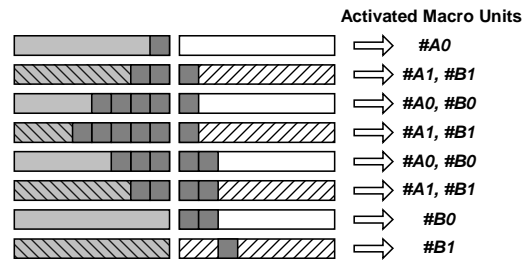


Fig. 6 : Selective Macro Activation

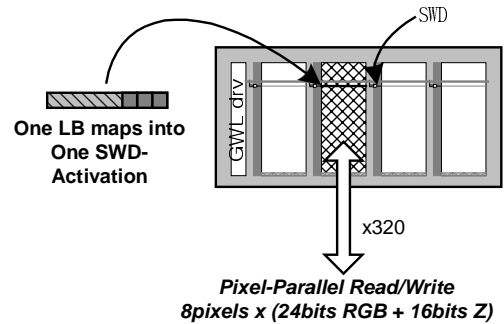


Fig. 7 : Partial Wordline Activation

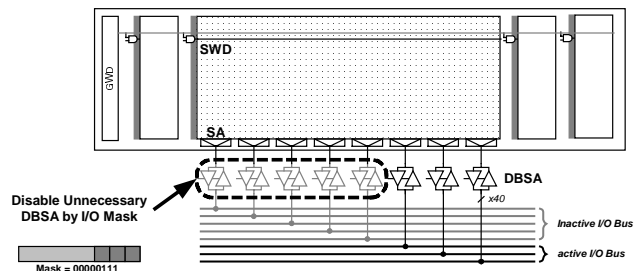


Fig. 8 : Partial I/O Activation

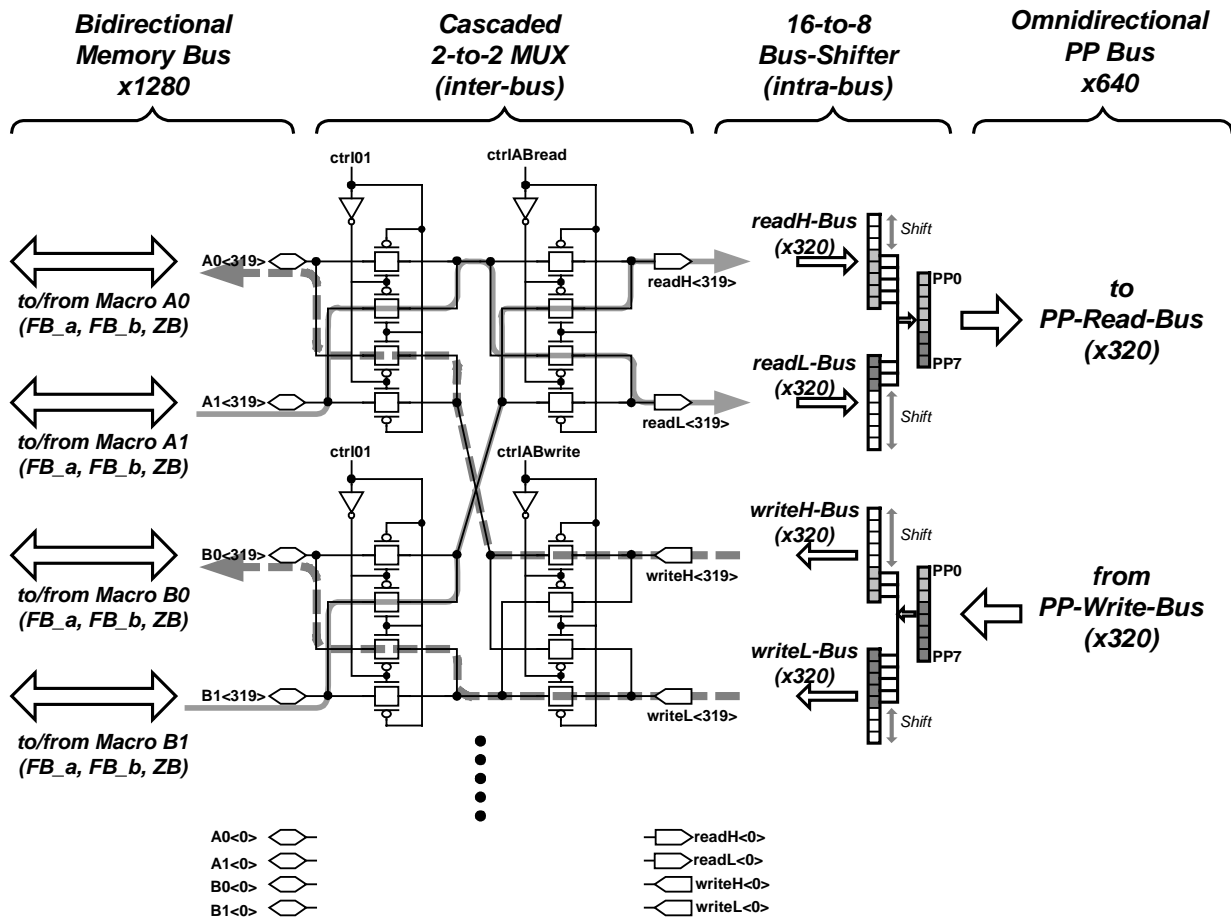


Fig. 9 : Circuit Diagram of R2bus

5. Implementation

Fig. 9 shows the circuit diagram of the proposed R2bus which contains cascaded 2-to-2 MUXs and 16-to-8 bus-shifters. The MUXs arbitrarily connect the bidirectional eDRAM bus to the omnidirectional PP bus by changing control signals (*ctrlI01*, *ctrlIABread*, *ctrlIABwrite*). And the bus-shifters assign pixel data to the corresponding PPs. This bus provides 3.2GByte/s bandwidth by accessing 1280bits of data simultaneously with read-modify-write pattern at 20MHz logic cycle. Besides, this R2bus can access eDRAM macros with three different modes for (a) normal rendering, (b) eDRAM test, and (c) eDRAM refresh as shown in fig. 10.

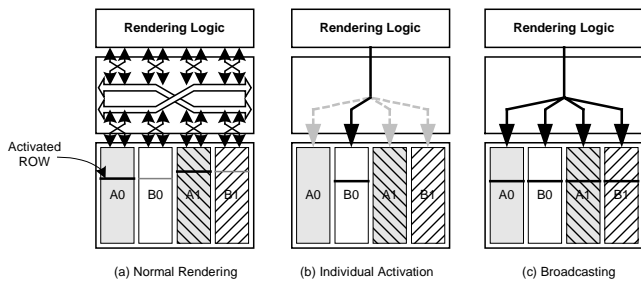


Fig. 10: Memory Access Modes of R2bus

Proposed E3GRE is integrated as the principal part of a PDA-chip and fig. 11 shows the 3D rendering flow in the chip [1]. After being pre-processed in an internal 32bit RISC

processor, 8x8-clipped polygon data, which are primitive components of 3D objects, are fed into the E3GRE to be displayed on an LCD screen. 3D rendering operations such as Gouraud shading, alpha-blending for transparency, depth comparison for hidden-surface removal, double-buffering for flicker-free animation, and direct video transfer are performed in E3GRE with fully utilizing the high bandwidth of eDRAM frame-buffer. All E3GRE blocks are designed, placed, and routed with full-custom method to optimally save the area and the power. The chip is fabricated by 0.18μm CMOS Embedded Memory Logic (EML) process with 3-poly 6-metal layers. 1.5V power supply is used for rendering logic, and 2.5V is applied to eDRAM macros. Fig. 12 shows its die-photo and table 1 summarizes its features.

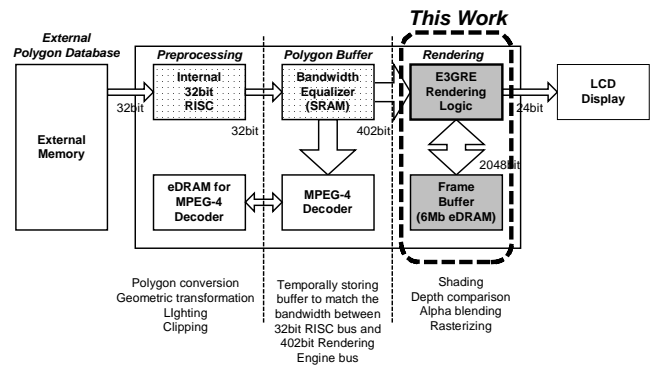


Fig. 11: 3D Rendering Flow in PDA-chip

6. Conclusion

An embedded 3D graphics rendering engine is designed to be integrated into a mobile PDA-chip with 6Mb eDRAM macros and 3.2GByte/s run-time reconfigurable bus. The proposed R2bus supports logically local frame-buffer architecture which reduces the area by 25% compared to the conventional local frame-buffers due to its high cell-efficiency. And polygon-dependent access to eDRAM macros with line-block mapping eliminates the unnecessary power consumption by 70% [fig. 13] while sustaining the read-modify-write data transaction. The E3GRE, which draws 2.22Mpolygons/s, was fabricated using 0.18 μ m CMOS EML technology and it shows 24mm² area and 120mW power consumption.

Acknowledgements

The authors thank to Hyundai System IC R&D Lab for chip fabrication and Se-Jeong Park of KAIST for gracious advice. This research was sponsored by System IC 2010 project of Korea Ministry of Science and Technology, and Ministry of Commerce, Industry and Energy.

References

- [1] Chi-Weon Yoon, et al, "A 80/20MHz 160mW Multimedia Processor integrated with Embedded DRAM, MPEG-4 Accelerator and 3D Rendering Engine for Mobile Applications," ISSCC, accepted for presentation, TA 9.2, 2001
- [2] Yong-Ha Park, et al, "A 7.1GB/s Low-Power 3D Rendering Engine in 2D Array Embedded Memory Logic CMOS," ISSCC, Dig of Tech. Papers, pp. 242-243, 2000
- [3] Yoshihara Aimoto, et al, "A 7.68GIPS 3.84GB/s 1W

Parallel Image-Processing RAM Integrating a 16Mb DRAM and 128 Processors," ISSCC, Dig of Tech. Papers, pp. 372-373, 1996

[4] Takao Watanabe, et al, "A Modular Architecture for a 6.4-Gbyte/s, 8Mb DRAM-Integrated Media Chip," Journal of Solid-State Circuits, pp. 635-641, May, 1997

[5] Kazunari Inoue, et al, "A 10Mb Frame Buffer Memory with Z-Compare and A-Blend Units," Journal of Solid-State Circuits, pp 1563-1568, Dec, 1996

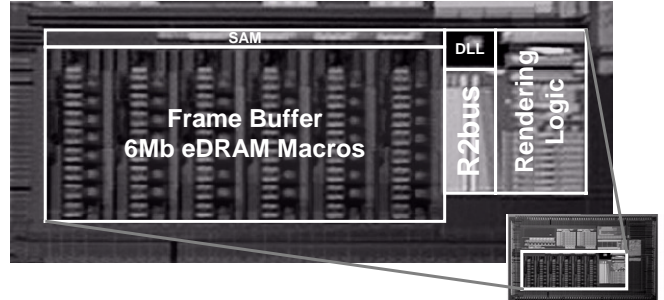


Fig. 12 : Die-Photo

Table 1 : E3GRE Features

Process	0.18mm CMOS EML with 3-poly 6-metal	
Power Consumption	120mW	
Area	24mm ²	
Power Supply	Rendering Logic	1.5V @ 20MHz
	eDRAM	2.5V @ 100MHz
Components	190,231 logic transistors 6Mb eDRAM Macros	
Data Rate	3.2GByte/s	
3D Rendering Features	2.22Mpolygons/s Gouraud Shading 16bit Depth-Comparison Alpha-Blending Double-Buffering Direct Video Transfer through SAM 24bit true-color on 256x256	

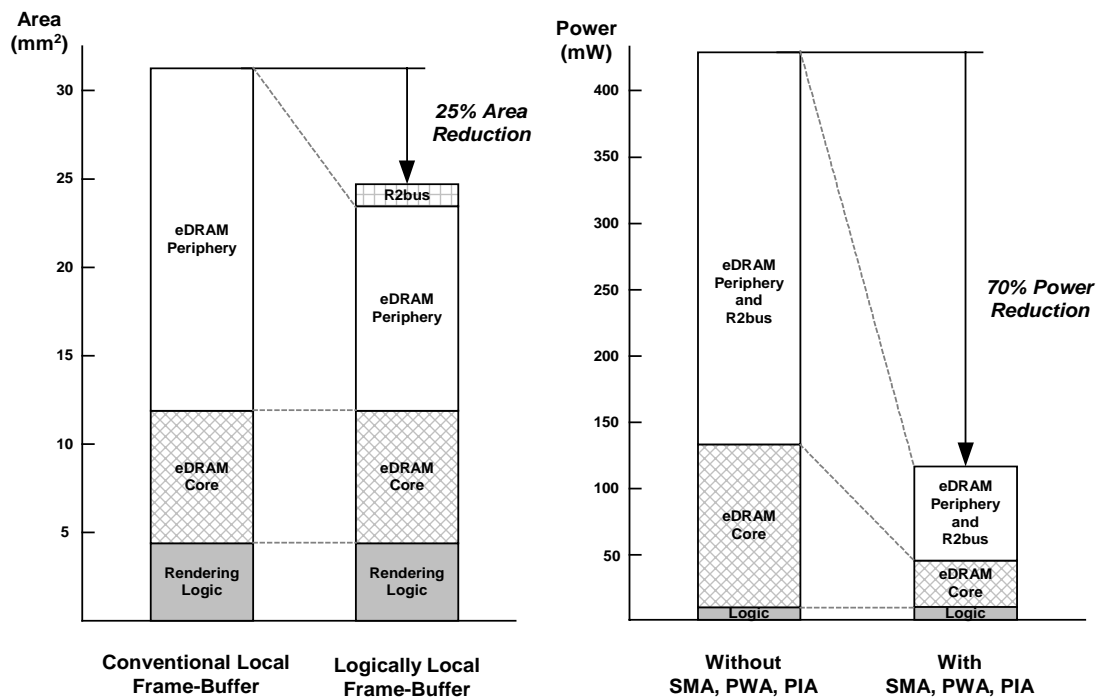


Fig. 13 : Area and Power Reduction