A 7.1-GB/s Low-Power Rendering Engine in 2-D Array-Embedded Memory Logic CMOS for Portable Multimedia System

Yong-Ha Park, Seon-Ho Han, Jung-Hwan Lee, and Hoi-Jun Yoo

***Abstract*—A single-chip rendering engine that consists of a DRAM frame buffer, a SRAM serial access memory, pixel/edge processor array and 32-b RISC core is proposed for low-power three-dimensional (3-D) graphics in portable systems. The main features are two-dimensional (2-D) hierarchical octet tree (HOT) array structure with bandwidth amplification, three dedicated network schemes, virtual page mapping, memory-coupled logic pipeline, low-power operation, 7.1-GB/s memory bandwidth, and 11.1-Mpolygon/s drawing speed. The 56-mm^2 prototype die integrating one edge processor, eight pixel processors, eight frame buffers, and a RISC core are fabricated using 0.35-μm CMOS embedded memory logic (EML) technology with four poly layers and three metal layers. The fabricated test chip, 590 mW at 100-MHz 3.3-V operation, is demonstrated with a host PC through a PCI bridge.**

***Index Terms*—Embedded memory, embedded logic, 3-D graphic rendering.**

I. Introduction

SYSTEM-ON-A-CHIP (SOC) will be the key technology in the twenty-first century semiconductor industry. Embedded memory logic (EML) is known to be a promising solution for SOC because it can achieve low power, high bandwidth, and small board area. These features are very important factors in the implementation of portable systems. Recently, portable devices have extended their multimedia application area to voice and two-dimensional (2-D) image processing in videophones, and even to the realization of a multimedia system on a chip [1]–[3]. Three-dimensional (3-D) graphics present attractive applications that may be included in the portable multimedia terminal such as a PDA or mini-game machine [1], [3].

Advances in VLSI technology have extremely accelerated 3-D graphics performance since the 1990s. High-performance graphic workstations generate real-time realistic scenes [4]. Widely used graphic accelerators enable 3-D graphic capabilities in PCs or game machines [5], [6]. Through EML technology, a DRAM frame buffer and a graphic accelerator are integrated into a single chip for possible use in notebook PCs or laptop computers [7]. Graphics performance of three application areas differs from each other in order to meet their constraints: space, power budget, and cost problems. In engineering workstations or PC graphic cards, high-performance frame buffers such as SRAM or synchronous graphic DRAM are widely used to meet their performance requirements [5]. In portable applications, an integration of the DRAM frame buffer is more attractive than off-chip solutions because of low-power high-bandwidth characteristics [8]–[10]. Typical rendering characteristics utilized in various application areas are summarized in Table I. A special hardware (H/W) accelerator is an essential requirement even in low-end graphics. This is because a drawing rate using only software library, 10K polygons/s, is insufficient for modern graphic applications [11].

There have been several EML approaches used in differing application areas. One is a general-purpose system that uses an advanced computer architecture and the other is an application-specific H/W that focuses on special enhanced features such as high bandwidth, low power, and small board area [8]–[10], [12], [13]. For general-purpose systems, they commonly integrate scalar processors, DRAM main memory, and SRAM cache memory on a single chip [12], [13]. But it is known that the "blind" integration of such components is insufficient to obtain performance improvement except for power and area savings due to the board size reduction. This is because a well-developed hierarchical memory system combining register, cache memory, and main memory hides the advantages coming from the extended memory bandwidth of EML. To overcome this drawback, vector units are incorporated for high-performance computing [12] or multiple processor–DRAM pairs are integrated on a single chip [13]. On the other hand, application-specific H/W accelerators using EML technology are developed for multimedia applications such as image processing or 3-D graphic processing [8]–[10]. They usually take advantage of parallelism to obtain the real-time multimedia data processing. For example, a single-instruction multiple-data (SIMD) array architecture that combines 128 processors and a 16-Mb DRAM has been developed for image processing [8]. A 3-D graphic pixel processor integrating a DRAM frame buffer converts read-modified-write (RMW) memory access to an atomic memory access [9], [14].

However, these previous EML approaches have several drawbacks for portable 3-D graphics because of their high power consumption, load imbalance, and large board area due to additional chips and interconnections. The proposed 3-D rendering

Manuscript received July 14, 2000; revised January 17, 2001. This work was supported by System IC 2010 Project of the Korea Ministry of Science and Technology, and Ministry of Commerce, Industry and Energy.

Y.-H. Park and H.-J. Yoo are with the Department of Electrical Engineering, Korea Advanced Institute of Science and Technology, Taejon 305-701, Korea (e-mail: yhpark@eeinfo.kaist.ac.kr).

S.-H. Han is with the Electronics and Telecommunications Research Institute, Taejon 305-350, Korea.

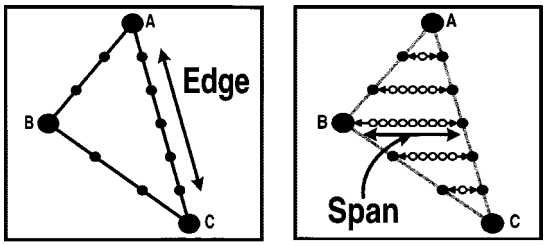
J.-H. Lee is with the System IC R & D Lab, Hyundai Electronics Industrial Company, Ltd., Kyoungki-do 467-701, Korea.

Publisher Item Identifier S 0018-9200(01)04127-0.

TABLE I
RENDERING CHARACTERISTICS

	EW/S or PC	NoteBook PC	Proposed
Frame Buffer (Typical)	Ext. 32~64 MB	Int. 6 MB	Int. 512KB
Polygon Size (Typical)	10×10	10×10	8×8
Pixel (Color/Depth)	32b / 32b	32b / 32b	24b / 16b
Shading	Smooth / Phong	Smooth	Smooth
Texture Mapping	Std.	Bilinear / Mipmap	To be considered
Advanced Features	Bumpmap, Fogging, ...	Fogging, ...	N.A.

(Std.: Anisotropic / Trilinear / Mipmap / Antialiasing)



(a) (b)

Fig. 1. Two-level hierarchical parallelism. (a) Polygon-level parallel operation (vertical shading). (b) Pixel-level parallel operation (horizontal shading).

engine resolves these problems by integrating a DRAM frame buffer, an SRAM serial access memory, a pixel/edge processor array, and a 32-b RISC core on a single chip. The main features of the proposed rendering engine are 2-D hierarchical octet tree (HOT) array structure with bandwidth amplification, three dedicated networks, virtual page mapping, a memory-coupled logic pipeline, and low-power operation. These features satisfy requirements for high bandwidth, low power, and small board area that are critical in portable 3-D graphic devices.

In this work, screen resolution and polygon size are assumed to be 256×256 pixels and smaller than 8×8 pixels, respectively. A polygon larger than 8×8 pixels is preclipped to multiple subpolygons smaller than 8×8 pixels at the preprocessing RISC. For small polygons, if the overall size of multiple neighbor polygons is smaller than 8×8 pixels, these polygons can be processed in parallel in single memory cycle. Four primitive rendering functions, smooth shading, depth comparison, alpha blending, and double buffering, are supported. Two shading operations are performed by two kinds of processor arrays, respectively. One is vertical shading at eight edge processors (EPs) and the other is horizontal shading at 8×8 pixel processors (PPs) array. All pixel information along a polygon edge is determined by vertical shading, as shown in Fig. 1(a). Next, all pixel information on eight spans is calculated in parallel from horizontal shading, as shown in Fig. 1(b). Both shading operations are based on asynchronous propagation of partial accumulation. After complete shading, RMW frame buffer (FB) access and pixel operations such as depth comparison and alpha blending take place in parallel across an 8×8 PP-FB array. It takes a single memory access cycle, nine clocks at 100 MHz including 30 ns precharge time, to perform horizontal shading and pixel operations over 8×8 pixels. Additionally, double buffering is supported in order to eliminate a screen flicker during moving image display.

The proposed rendering architecture will be introduced in Section II. The implementation of processing elements and memory units will be described in Section III. Fabrication and measurement will be explained in Section IV, and conclusions will follow in Section V.

II. ARCHITECTURAL FEATURES

A. 2-D Hierarchical Octet Tree (HOT) Array Structure with Bandwidth Amplification

Efficient parallel processing is critical to realizing real-time multimedia data processing [3]. A flat one-dimensional (1-D) array processor is usually adopted in 2-D image processing such as FFT, DCT, IDCT, and convolution [8], [18]. However, rendering performance integrating a 1-D array processor and embedded DRAM frame buffer is similar to that of the software (S/W) library solution [9], [11]. This is because the utilization of two-level hierarchical parallelism is insufficient even though huge parallel-pixel operation and atomic RMW memory access can be achieved [9].

The proposed 2-D HOT array structure consists of a preprocessing RISC core, bus matching queue, edge processors (EPs), pixel processors (PPs), DRAM frame buffers (FBs), and SRAM serial access memories (SAMs). It has a two-level octet tree structure that is made of a single master processor and eight slave processors at each level, as shown in Fig. 2. A set of one preprocessor and eight EPs makes the first-octet processing layer. Again, eight sets of EPs and eight PPs make the second-octet processing layer. These two layers support a two-level parallelism, coarse polygon parallelism at the first processing layer and fine pixel parallelism at the second processing layer. The coarse operation determines the polygon edge information and transfers it to the next layers. The fine pixel operation calculates the inner pixel information and performs pixel and RMW memory operations at the second processing layer. Both hierarchical operations cooperate together to construct a complete rendering pipeline including shading operation, pixel operations, RMW memory operation, and screen refresh operation.

Bandwidth amplification is obtained from the bus configuration of the 2-D HOT array structure, as shown in Fig. 2, after the 32-b RISC preprocessor supplies data and commands to the bus

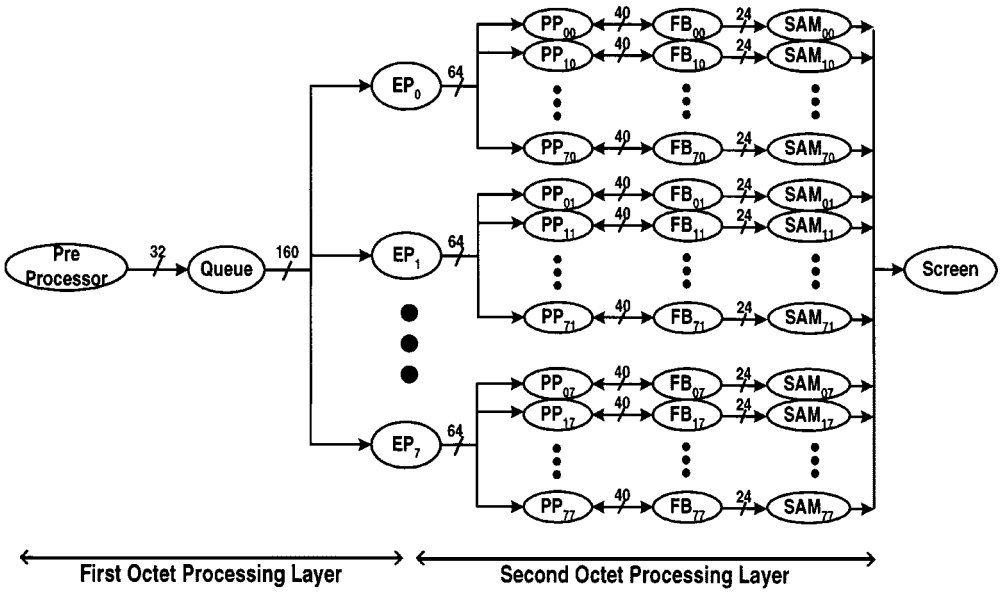


Fig. 2. Proposed 2-D hierarchical octet tree (HOT) structure.

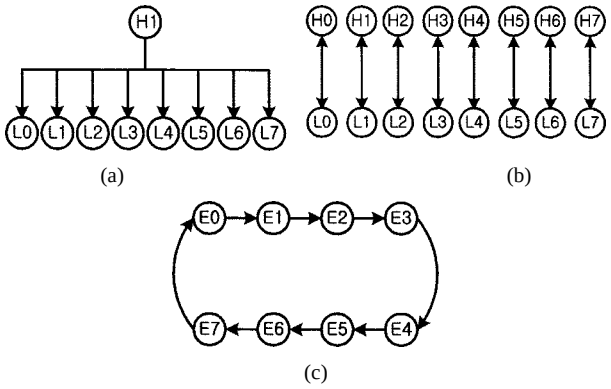


Fig. 3. Three network schemes. (a) Unidirectional broadcasting mode(UBM). (b) Bidirectional point-to-point mode (BPM). (c) Unidirectional ring mode (URM).

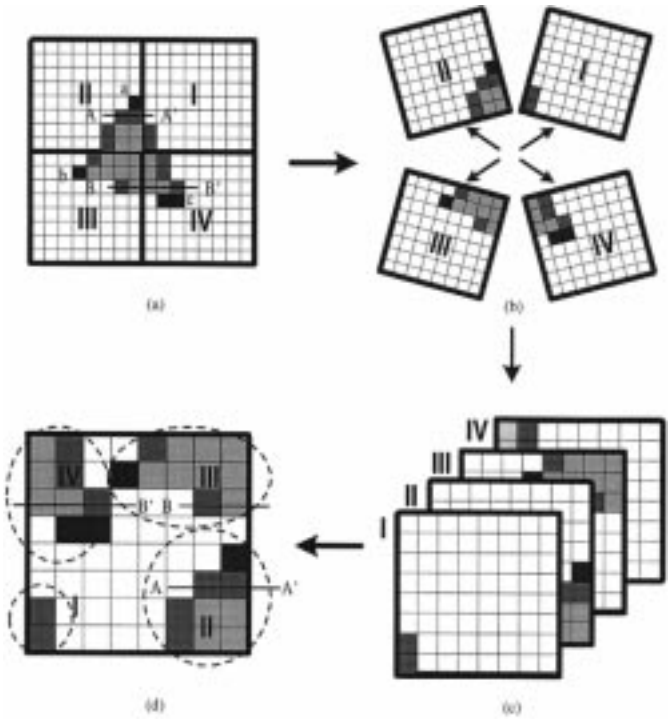


Fig. 4. Example of virtual page mapping. (a)–(c) 8×8 pixels on a screen. (d) 2-D HOT 8×8 array processor assigned to corresponding 8×8 pixels.

matching queue through the 32-b width bus. The queue extends the bus width from 32 b up to 160 b in order to feed them into eight EPs. Each EP is a master to its own eight PPs, connected by a 64-b bus. There are eight groups made of an EP and eight PPs. Each PP directly interfaces with the corresponding FB through the 40-b RMW data bus. Finally, 2560 b of data can be accessed by overall 64 RMW operations within single memory cycle, nine clocks at 100 MHz or 7.1 GB/s frame buffer access bandwidth. Bandwidth amplification with 80 times gain is achieved. This is difficult to implement in a space-limited PCB but can be obtained in the HOT architecture using EML.

B. Transactions of Each Processing Element

For efficient parallel processing, the three dedicated network schemes of Fig. 3(a)–(c) cooperate to overcome the communication bottleneck coming from the conventional bus scheme. Unidirectional broadcast mode (UBM), shown in Fig. 3(a), is applied to the data transaction from one master processor to its eight slave processors, such as between a preprocessor and eight EPs or between an EP and eight PPs. Bidirectional point-to-point mode (BPM), shown in Fig. 3(b), performs a RMW transaction for depth comparison and alpha blending operation at each PP-FB pair. Unidirectional ring mode (URM), shown in Fig. 3(c), is applied among EPs or PPs themselves. It connects each EP with two neighbor EPs, forming a ring topology in order to propagate the partial accumulation for vertical shading. For horizontal shading, eight PPs use URM in a similar way to EP interconnection. These three network schemes guarantee two-level parallelism and bus conflict-free communication for both data processing and memory reference.

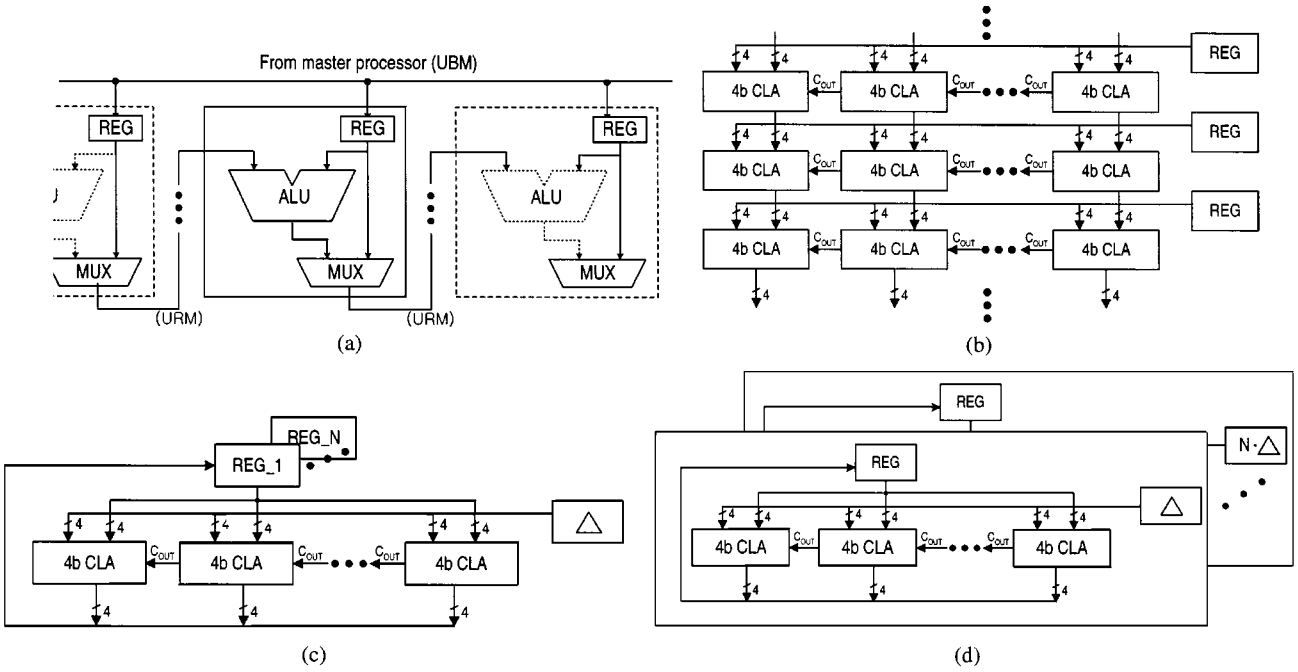


Fig. 5. (a) IU block diagram. (b) Asynchronous propagation scheme using an array of 4-b carry–look-ahead (CLA) adder. (c) Synchronous sequential scheme. (d) Synchronous parallel scheme.

C. Virtual Page Mapping (VPM)

In 3-D computer graphic rendering, frame buffer access characteristics show the spatial locality rather than the temporal locality [14], [15]. This is because the basic primitives for 3-D graphics are triangles or rectangles that have spatial locality on a screen. Each pixel inside an object or a polygon is accessed only once until the object or the polygon is completely drawn. Unless the mapping from a pixel on a screen to the frame buffer is optimized, the spatial locality on the frame buffer may be reduced since pixels inside a polygon are stored in multiple rows [15]. This causes bandwidth degradation for both the RWM memory operation and the screen refresh operation [14]. Horizontal page mapping has an advantage in the screen refresh operation by using a fast page mode access, but suffers from a high memory row miss penalty during 3-D rendering operations. Although tile mapping (or rectangular page mapping) effectively trades off rendering bandwidth with screen refresh bandwidth, it needs four different memory row cycles when a polygon straddles four tiles, as shown in Fig. 4(a). This is because each pixel group on four tiles is stored in a different row. Here, "tile" refers to a rectangular pixel array on screen stored in the frame buffer with the same row address.

For a polygon clipped smaller than 8×8 pixels at the preprocessing RISC, virtual page mapping guarantees complete shading, pixel operations and RMW memory operation within a single memory cycle time, nine clocks at 100 MHz, without a memory row miss wherever the polygon is located on a screen. This is because all pixels inside a polygon smaller than 8×8 pixel array are mapped to the 2-D HOT 8×8 PP-FB array on a one-to-one basis, an operation which can be parallel. A pixel located in screen coordinate $(X[7{:}0], Y[7{:}0])$ is assigned to EP_n, PP_{mn}, and FB_{mn} that satisfy (1a) and (1b), where $X[7{:}0]$ and $Y[7{:}0]$ are horizontal and vertical screen coordinates of each pixel, respectively. m and n are horizontal and vertical indices of the array processor as shown in Fig. 2, respectively. Row and column addresses of (2a) and (2b) are used for each PP to access its own FB.

$$m = X \bmod 8 = X[2:0] \tag{1a}$$

$$n = Y \bmod 8 = Y[2:0] \tag{1b}$$

$$\text{Row address} = \left\{\frac{Y}{8}\right\} = Y[7{:}3] \tag{2a}$$

$$\text{Column address} = \left\{\frac{X}{8}\right\} = X[7{:}3] \tag{2b}$$

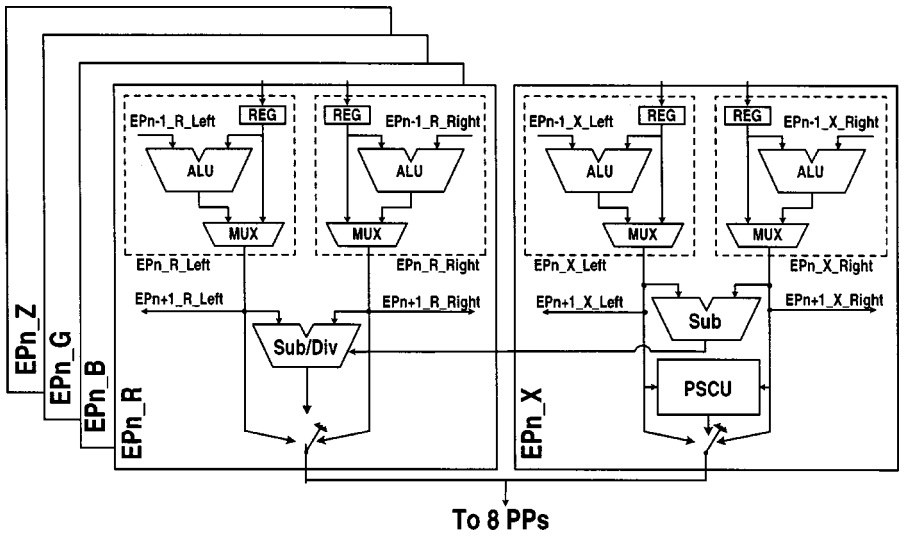
where $\{\}$ is the Gaussian operator that truncates the fractional part and takes only the integer part.

For example, assume that a polygon overlaps four different tiles, I, II, III, and IV, as shown in Fig. 4(a). We can divide the polygon into four pixel groups on each tile and rearrange them as shown in Fig. 4(b) and (c), respectively. Then we can successfully assign all pixels on four different tiles to the 2-D HOT 8×8 array where all pixels inside a polygon can be processed in parallel as shown in Fig. 4(d). Therefore, virtual page mapping achieves maximum utilization of parallelism and spatial locality while preserving the screen refresh mechanism of horizontal page mapping.

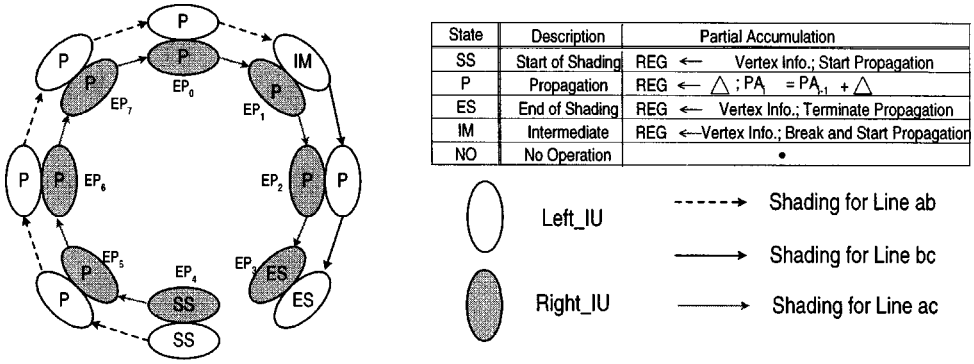
III. Design and Implementation

A. Interpolation Unit (IU) Using Asynchronous Propagation

Asynchronous propagation is applied to both vertical and horizontal shading at EPs and PPs, respectively. Fig. 5(a) shows the block diagram of an interpolation unit (IU), a basic unit for asynchronous propagation. IU has two input ports and one output port. One of the input ports is connected to its master



(a)



State	Description	Partial Accumulation
SS	Start of Shading	REG ← Vertex Info.; Start Propagation
P	Propagation	REG ← △ ; PA_i = PA_{i-1} + △
ES	End of Shading	REG ← Vertex Info.; Terminate Propagation
IM	Intermediate	REG ←Vertex Info.; Break and Start Propagation
NO	No Operation	•

(b)

Fig. 6. (a) EP block diagram. (b) Vertical shading example for the vertex a, b, and c of Fig. 4(a).

processor (either preprocessor or EP) in order to receive data through the UBM interface. The other input port is connected to the output port of a neighbor EP so that a partial accumulation (PA) can be asynchronously propagated through the URM interface. There are three kinds of IU operations, starting propagation, generating and propagating PA, and terminating propagation. In starting propagation, pixel data stored at internal register through UBM is propagated to the next IU via MUX without accumulating. In generating and propagating PA, incremental data (Δ) received from UBM is stored at internal register. IU generates and propagates PA_i to the next IU by accumulating Δ and previous PA_{i-1} supplied from URM. In terminating propagation, pixel data from UBM is only stored at the register without any accumulation and propagation.

The array of 4-b carry–look-ahead (CLA) adder is used for generating PA at each IU, and its PA is delivered to next IU as shown in Fig. 5(b). It achieves high speed and small peak power consumption compared to synchronous sequential and synchronous parallel accumulation, respectively. A synchronous sequential accumulation, as shown in Fig. 5(c), is suitable for deeply pipelined pixel operations rather than parallel pixel operations because its long latency of $N \cdot (\tau_{\mathrm{REG}} + M \cdot \tau_{4\mathrm{bCLA}})$ prohibits the next following parallel operations where τ_{REG}, $\tau_{4\mathrm{bCLA}}$, M, and N are a critical time for the register, a critical time for the 4-b CLA adder, a number of 4-b CLA stage at each IU, and a number of propagation stage, respectively. Despite high-speed operation of $\tau_{\mathrm{REG}} + M \cdot \tau_{4\mathrm{bCLA}}$, a synchronous parallel accumulation, as shown in Fig. 5(d), suffers from two overheads. One is generating a multiple of differential values from Δ to $N \cdot \Delta$. The other is high instantaneous peak power consumption during parallel accumulation and latch operations of internal register. The proposed asynchronous propagation is well matched to the 2-D HOT array and achieves a balanced processing rate between two kinds of shading operations and their next following operations. It is because either vertical or horizontal shading over 8×8 pixel array is completed within only $\tau_{\mathrm{REG}} + (N + M - 1) \cdot \tau_{4\mathrm{bCLA}}$ so that the next following operations of span information setup and pixel operations can be performed in parallel at EP's and PPs-FB's array, respectively.

B. Edge Processor (EP)

The edge processor EP_n contains 5 SIMD units of EP_n_R, EP_n_G, EP_n_B, EP_n_Z and EP_n_X as shown in Fig. 6(a) where n is the index of EP defined (1b). EP_n_R, EP_n_G, and EP_n_B are for three color information (R, G, B) processing. EP_n_Z and EP_n_X are for two coordinate information (X, Z) processing. Two IUs at each SIMD unit, Left_IU and Right_IU, perform two vertical shading operations along both the left and right edges of a polygon by using asynchronous propagation,

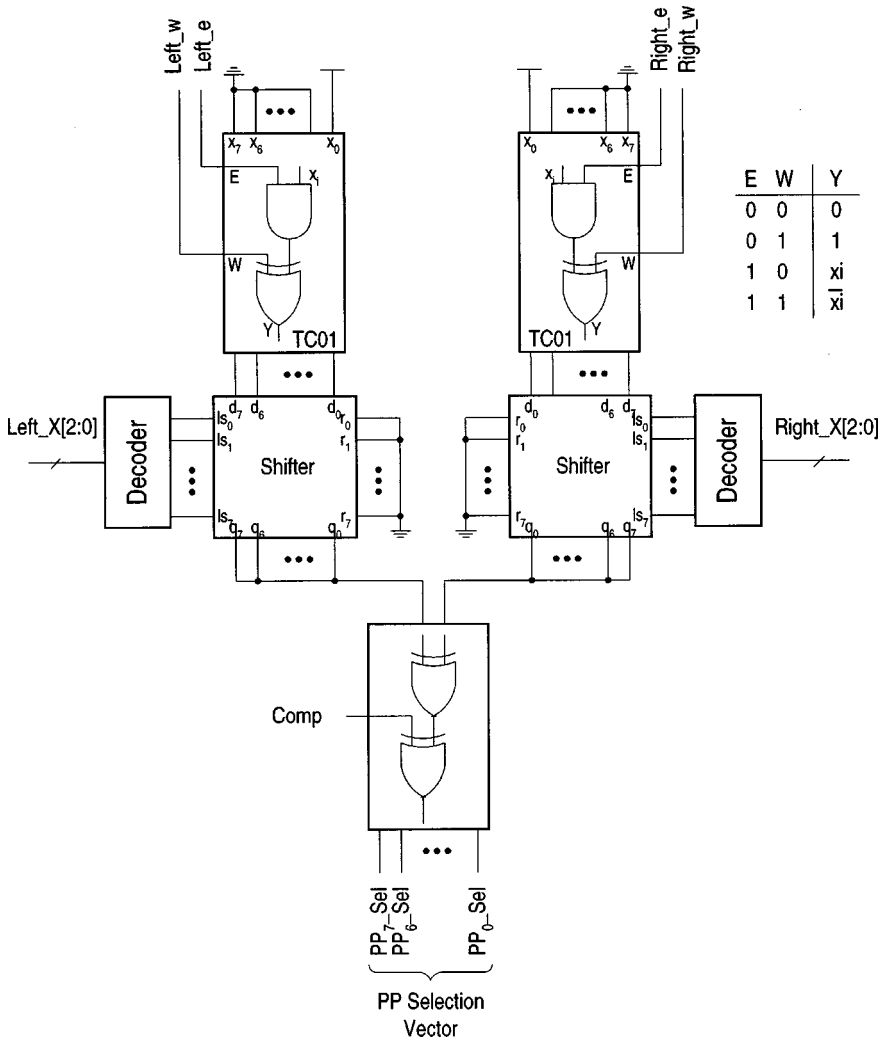
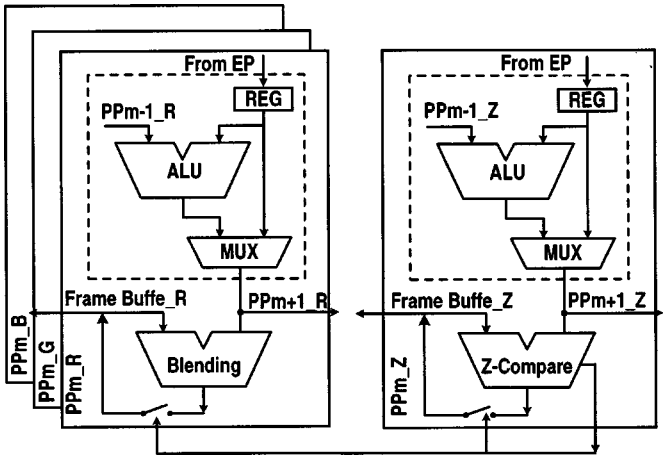


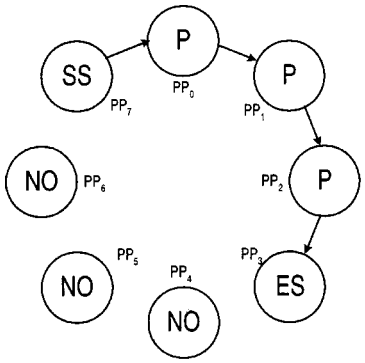
Fig. 7. (a) Simplified pixel processor selection unit (PSCU) and control signals. (b) Control timing diagram.

as shown in the example of Fig. 6(b). A command stored at the queue is broadcast in order to activate the selected EP. Then the vertex information on points a, b, and c of Fig. 4(a) is loaded into each internal register of selected EPs that satisfy (1b) for three vertices. These EPs are assigned one of three statuses such as start of shading (SS), end of shading (ES), and intermediate (IM). The line information on lines ab, bc, and ac of Fig. 4(a) are loaded into the corresponding internal register of EPs with propagation status (P). After complete vertical shading, EPs set up their own span information packet including two kinds of information. One is pixel information (R, G, B, Z) about both end pixels of each span obtained from vertical asynchronous propagations. The other is differential information $(\Delta \mathrm{R}, \Delta \mathrm{G}, \Delta \mathrm{B}, \Delta Z)$ between them. To obtain a differential value (Δ), a nonrestoring array divider is used [16].

The pixel processor selection unit generates an 8-b PP selection vector in order to transfer a span information packet to corresponding PPs, as shown in Fig. 7(a). Two 8-b TC01 blocks generate two preliminary selection vectors by using four control signals, Left_e, Left_w, Right_e, and Right_w. Two horizontal screen coordinates at both end pixels of a span, Left_$X[7{:}0]$ and Right_$X[7{:}0]$, are partially decoded in order to determine each left-shift amount, respectively. The PP selection vector is obtained from bit-wise XOR operation over two left-shifted preliminary selection vectors. When Left_$X[2{:}0]$ is larger than Right_$X[2{:}0]$, this PP selection vector is bit-wise inverted by ac-



(a)



State	Description	Partial Accumulation
SS	Start of Shading	REG ← Left Pixel Info.; Start Propagation
P	Propagation	REG ← △ ; $PA_i = PA_{i-1} + \triangle$
ES	End of Shading	REG ←Right Pixel Info.; Terminate Propagation
NO	No Operation	

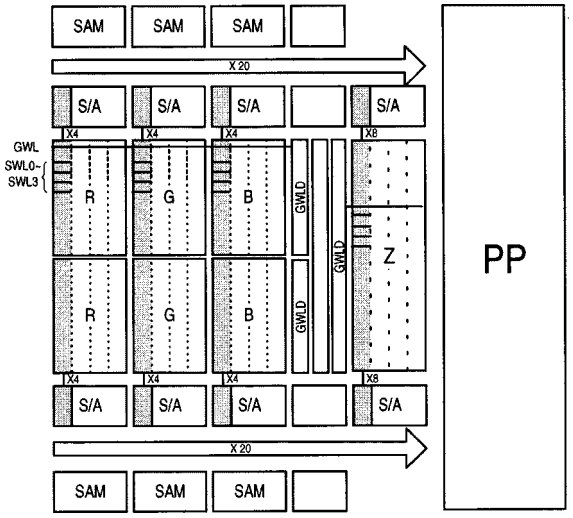
(b)

Fig. 8. (a) Pixel processor block diagram. (b) Horizontal shading example for the span B–B′ of Fig. 4(a).

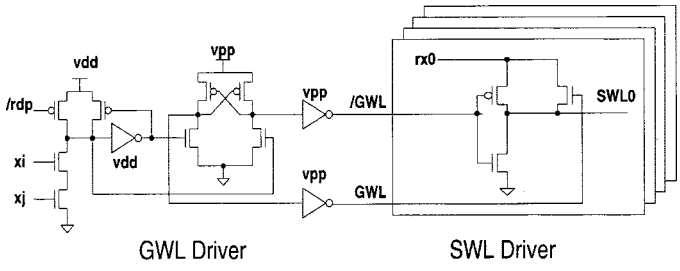
tivating the Comp signal. The timing diagram of pixel processor selection unit is shown in Fig. 7(b).

C. Pixel Processor (PP)

A pixel processor comprises four SIMD units of PPm_R, PPm_G, PPm_B, and PPm_Z, as shown in Fig. 8(a), where m is the index of PP defined in (1a). Each SIMD unit consists of a single IU, a memory control unit, a blending unit for color information, and a compare unit for depth information. After a span information packet is transferred from an EP to its eight PPs, each PP is assigned to one of four propagation statuses, as shown in Fig. 8(b). Asynchronous propagation continues in all PPs with propagation state (P) between start of shading (SS) and end of shading (ES). Fig. 8(b) shows a horizontal shading example for the span B–B′ of Fig. 4(a). PP with SS status loads left pixel information of the span and transfers it to the next PP without accumulation. PP with ES status loads right pixel information of the span and terminates propagation. PPs with P status load differential information (Δ) of the span and perform both accumulation and propagation.



(a)



(b)

Fig. 9. (a) Frame buffer (FB) structure. (b) Global wordline and subwordline drivers.

There are two different types in this propagation. One is direct propagation and the other is turn-around propagation. In the direct propagation, Left_X[2:0] is smaller than Right_X[2:0]. For example, in the span A–A′, as shown in Fig. 4(a), all pixels on the span are mapped to their corresponding PP's array on A–A′ as shown in Fig. 4(d). All PPs access their own FB with the same column address of Left_X[7:3] (or Right_X[7:3]) as defined in (2b). In the turn-around propagation, Left_X[2:0] is larger than Right_X[2:0]. For example, in the span B–B′, as shown in Fig. 4(a), there are two pixel groups belong to different tiles. Each group is cross assigned to their PPs although the pixel sequence inside a group is preserved, as shown in processor array on B–B′ of Fig. 4(d). Two kinds of column addresses, either Left_X[7:3] or Right_X[7:3], are used for each pixel group to access their FB.

After complete horizontal shading, each PP simultaneously performs five-level alpha blending and 16-b depth comparison in the sequence of R, G, B, and Z through RMW memory access. Memory control signals, including the address, data, read, write, update and ras signals, are generated at each PP.

D. Frame Buffer (FB) and Serial Access Memory (SAM)

Fig. 9(a) shows the structure of a 64-kb DRAM FB. It is composed of four memory blocks storing R, G, B, and Z components. Each block has four segments for independent activation using global wordline and subwordline drivers as shown in Fig. 9(b). Each segment has its own 4-kb cell array, sense amplifier, and subwordline driver. The SAM is attached to only the

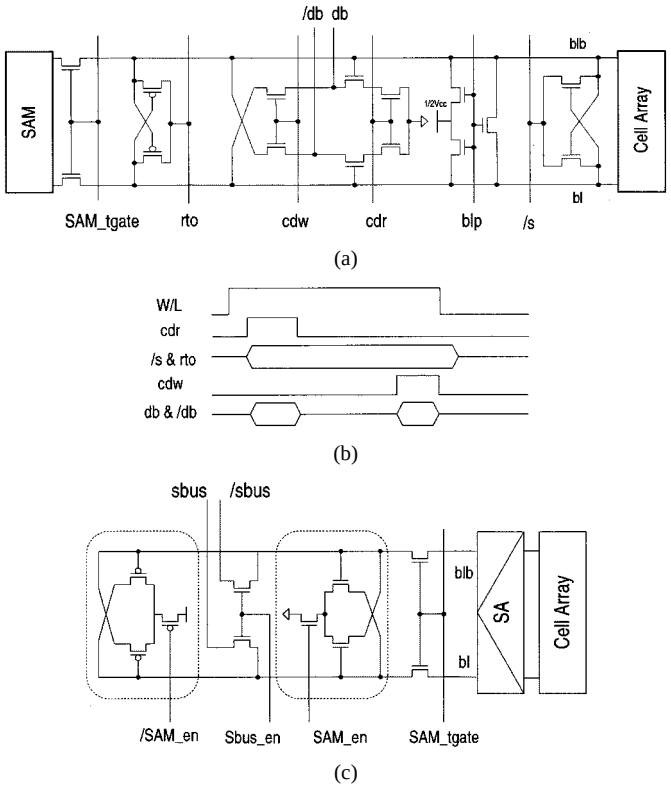


Fig. 10. (a) Sense amplifier. (b) Its timing diagram in 3-D rendering mode. (c) Serial access memory (SAM).

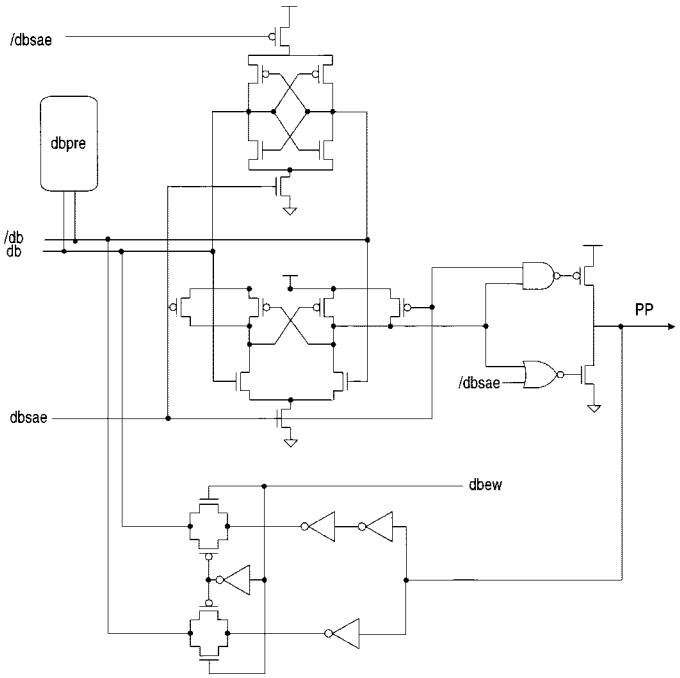


Fig. 11. Two-stage DBSA and writing circuits.

R, G, and B blocks because they need to scan out each pixel value to the screen. The R, G, and B blocks have a pair of cell arrays to support a double buffering in order to eliminate screen flicker while a 3-D moving image is displayed. The depth block has its own wordline decoder and wordline driver because the same memory cell in Z block should be activated even if two memory addresses caused by double buffering are alternately accessed in the R, G, and B blocks. This scheme consumes less area than a duplicated Z block of the paper [9].

There are four FB operation modes controlled by the column path, including bitline sense amplifier (BLSA), data bus (db)-line sense amplifier (DBSA) and SAM; a DRAM refresh, a 3-D rendering, a SAM transfer, and initialization. Fig. 10(a) shows a direct sense amplifier (SA) using shared db-line for read operation as well as write operation. This scheme reduces the number of db-line pairs to half at the PP-FB. In DRAM refresh mode, only /s and rto signals activate a cross-coupled SA to restore the cell capacitor. In 3-D rendering mode, fast direct bitline sensing develops the signal on a db-line pair when cdr is activated. Cell data is either updated or not according to the result of depth comparison and alpha blending when cdw is activated. Fig. 10(b) shows a BLSA timing diagram in 3-D rendering mode. In SAM transfer mode, a SAM_tgate of Fig. 10(c) is activated after the cross-coupled SA completes bitline sensing. In this operation, a complementary SAM_en signal is disabled in order to help data transfer even when the previous latched data is opposite to that of current sensing data. A transfer operation is completed when complementary SAM_en signals are activated and a SAM_tgate is closed. Pixel data, stored in a SAM, can be serially read out to a screen or video card when the Sbus_en signal is selected. In initialization mode, PP directly writes cell data in order to clear the depth buffer or paint a 2-D background image without read-modify operation. For PP interface, a 2-stage DBSA and writing circuits, as shown in Fig. 11, is enabled in both 3-D rendering and initialization mode. Memory test is achieved by using SAM read operation after both the initialization step and the SAM transfer step.

There are two kinds of FB access. One is a local reference and the other is a global reference. The local reference is 40-b RMW access based on BPM between a PP and its local FB. Sixty-four local references across an 8×8 PP-FB array provide a wide bandwidth for rendering operation even though one pixel data is accessed in each local memory reference. This is because parallel processing combined with virtual page mapping is applied rather than pipelined processing combined with the conventional page mode access for PPs-FB's interface. The global reference is 24-b read-only access from SAM to screen. SAM separates the local memory reference from the global reference.

Use of a wide DRAM page to increase the hit ratio makes the power budget worse [17]. In particular, the power consumption of an EML chip is critical in determining the DRAM cell refresh period that decreases at high temperature. Two low-power schemes, partial segment activation (PSA) and sequential block activation (SBA), reduce the power consumption of each nodal FB to 20 mW. Partial segment activation minimizes the power consumption during sense amplifier operation because only the subwordline and the sense amplifier array inside a target segment are activated [17]. The power consumption of the FB is analyzed in Fig. 12(a). The nodal power consumption of the FB reduces to 37.6% of the conventional scheme that simultaneously activates four segments, 256 cells. Another power reduction comes from sequential RMW operation in the sequence of

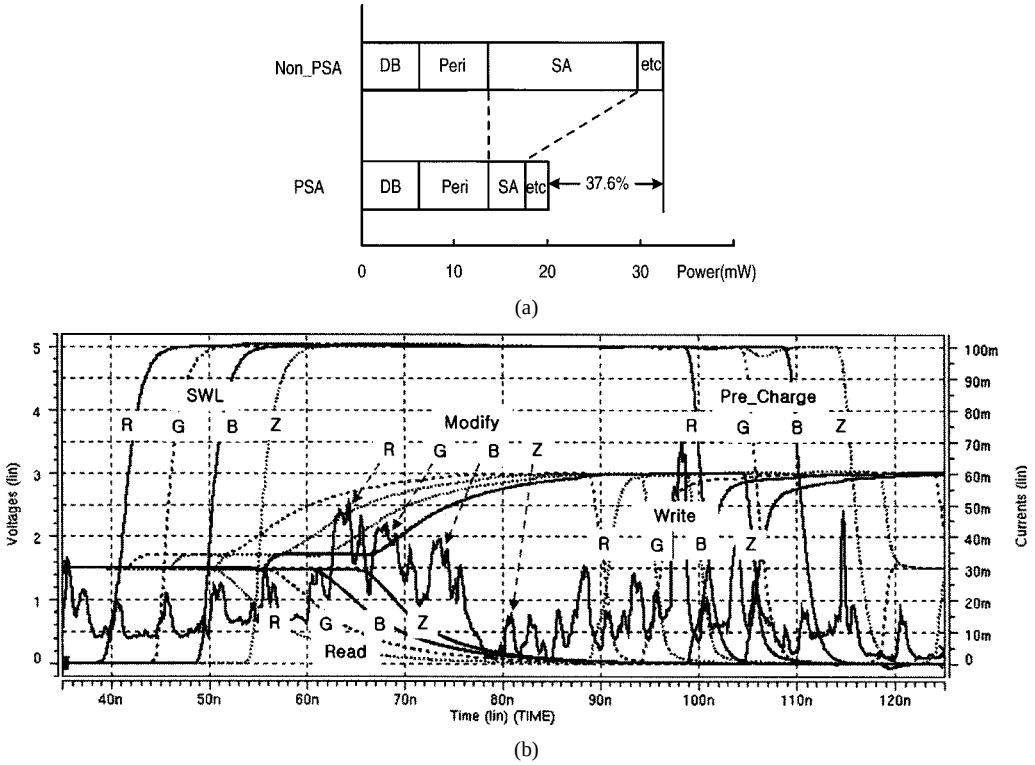


Fig. 12. (a) Power consumption of PSA versus non-PSA. (b) Simulation results of current flow through PP and FB using sequential block activation.

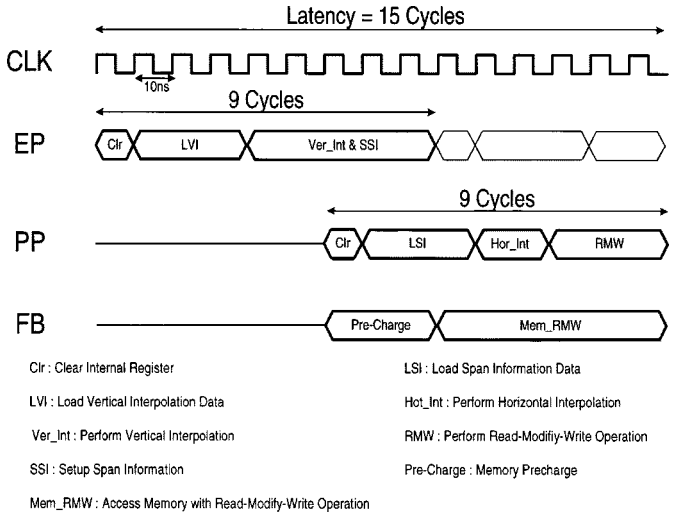


Fig. 13. Typical pipeline stage including EP, PP, and FB operation.

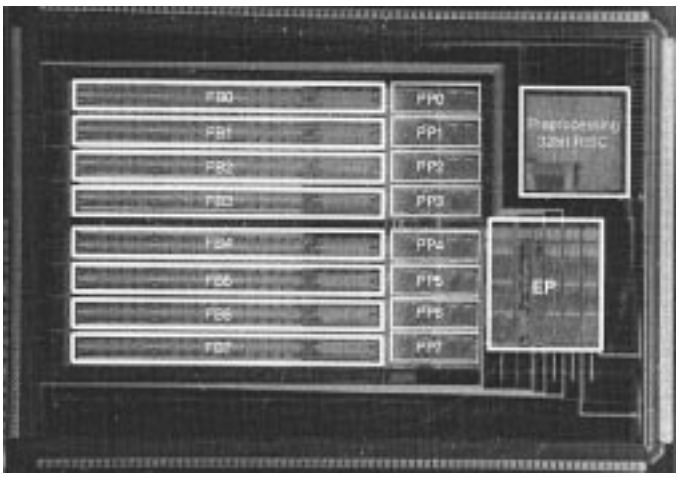
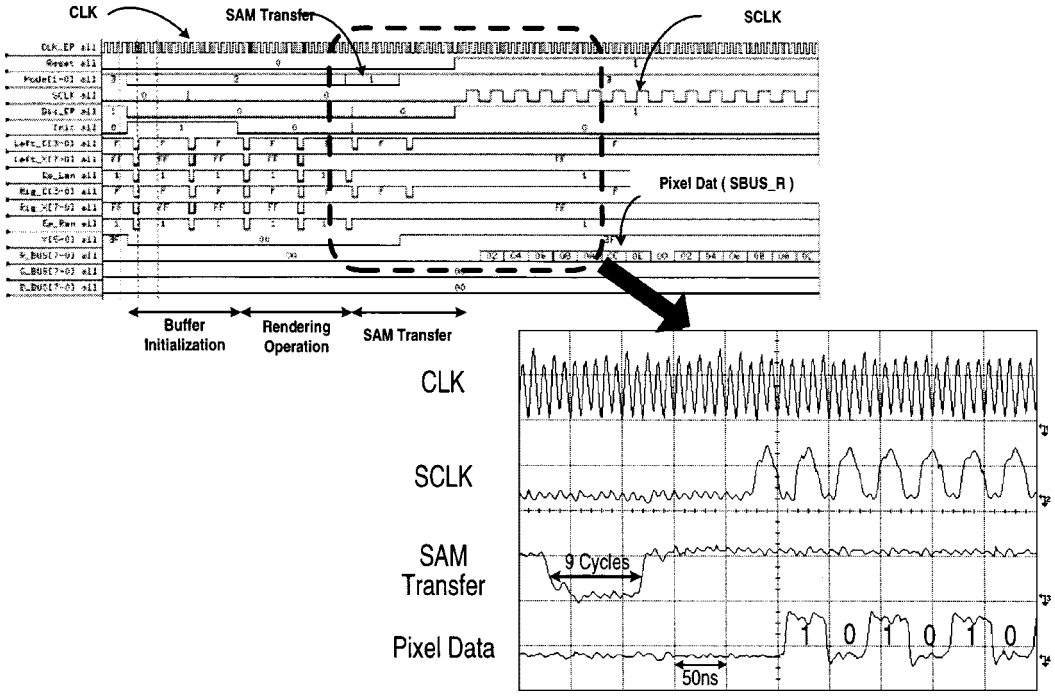


Fig. 14. Test chip microphotograph.

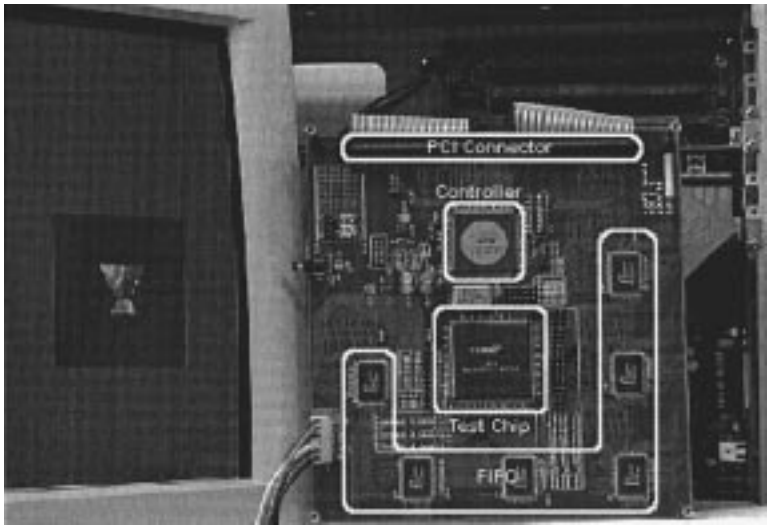
R, G, B, and *Z* block like a time-division multiplexed operation. Theoretically, the peak power consumption of SBA can be reduced to a maximum 75% that of simultaneous operation. The simulation results of operating current, including PP–FB operation using SBA, are shown in Fig. 12(b).

E. Memory-Coupled Logic Pipeline Operation

Typical pipeline operation of 2-D HOT array is tightly coupled with the FB memory cycle as shown in Fig. 13. It is composed of two partially overlapped nine-clock-latency stages with three shared clocks at 100 MHz; the front pipeline stage is for coarse polygon parallelism at EP and the back pipeline stage is for fine pixel parallelism between PP and FB. New polygon data can enter into the front pipeline stage in every nine clock cycles and performs the vertical shading and the span information setup. At the back pipeline stage, a tightly coupled PP–FB calculates all pixel values from the span information packet and performs pixel operations. Both pipeline stages are synchronized with the RMW memory operation that has 60-ns access time and 30-ns precharge time. They perform complete shading and pixel operations within every nine-clock-latency pipeline stage at 100 MHz. This results in 11.1M polygon/s drawing speed. Other memory operations such as SAM transfer, DRAM refresh, and initialization are also consistent with two nine-clock-latency pipeline stages, as well.



(a)



(b)

Fig. 15. (a) Measured waveforms. (b) Test board photograph.

IV. Fabrication and Measurement

Assuming 256×256 pixels resolution with 24-b true-color double buffer and 16-b depth buffer, the frame buffer contains 4-Mb DRAM and 6-kb SAM. A test chip including 1/8 of the proposed architecture is fabricated using 0.35-μm CMOS EML process with four poly layers and three metal layers. It integrates one EP, eight PPs, eight FBs, and eight SAMs with the HOT architecture. For flexible preprocessing, a 32-b RISC core is fabricated on the same die. A microphotograph of the test chip, 56 mm^2, including the 32-b RISC core, is shown in Fig. 14. The test chip characteristics are summarized in Table II. The digital waveforms measured by logic analyzer show four functional operations such as frame buffer initialization, 3-D rendering, SAM transfer, and pixel output, as shown in Fig. 15(a). The detail waveforms, including a 100-MHz system clock, 25-MHz SAM clock, SAM transfer command, and pixel output stream, are also shown in Fig. 15(a). After nine clock cycles for SAM transfer command in the front pipeline stage and an additional six clock cycles for memory operation in the back pipeline stage, the pixel data stream comes out synchronized with the SAM clock. For the overall functional test, the PCI interface unit, first-in-first-out (FIFO) external memory, and simple FPGA controller are assembled in the same board as shown in Fig. 15(b). During dump operation from the host PC to the test board, the FIFOs store input polygon information and commands. When a rendering operation is finished, the final pixel data are also stored at

TABLE II
TEST CHIP CHARACTERISTICS

	EP	PP	FB
Technology	0.35μm CMOS EML, 4polys 3metals		
Area	5.3mm^2	0.9mm^2	3.7mm^2
Clock	100MHz	100MHz	100MHz/33MHz
Supply Voltage	3.3V	3.3V	3.3V/5V
Power Consumption	150mW	35mW	20mW
Organization	1EP	8PPs	8FBs/8SAMs
Package	304 RQFP		
Total Power	590mW (Worst Case)		

the FIFO and transferred to the video card of the host PC and finally displayed on a screen.

V. CONCLUSION

Two-dimensional HOT array structure with bandwidth amplification and virtual page mapping are proposed to support two-level parallel rendering and the frame buffer access without page miss, respectively. Two low-power schemes, PSA and SBA, are applied for power reduction of the FB array. Three network schemes, UBM, URM, and BPM, guarantee inter processor communication without bus conflict. The proposed rendering engine achieves 7.1-GB/s memory bandwidth and 11.1M polygon/s drawing speed at 100-MHz 3.3-V operation. A test chip including 1/8 of the proposed engine is successfully fabricated by 0.35-μm CMOS EML technology and demonstrated in combination with a host PC and PCI-bridge. The multimedia system on a chip using EML technology gives a clear solution for portable 3-D graphic system under the constraints of low power, small area, and high bandwidth.

ACKNOWLEDGMENT

The authors would like to thank J.-S. Kim, S.-J. Lee, J.-H. Kook, J.-W. Lim, and R. Woo for their helpful assistance. The authors would also like to thank Dr. J.-H. Lee, S.-H. Han, and K.-J. Lee for their CAD assistance.

REFERENCES

[1] Y. H. Park *et al.*, "A 7.1 GB/s low-power 3-D rendering engine in 2-D array-embedded memory logic CMOS," in *ISSCC Dig. Tech. Papers*, 2000, pp. 242–243.
[2] T. Nishikawa *et al.*, "A 60-MHz 240-mW MPEG-4 video-phone LSI with 16-Mb embedded DRAM," in *ISSCC Dig. Tech. Papers*, 2000, pp. 230–231.
[3] I. Kuroda and T. Nishitani, "Multimedia processor," *Proc. IEEE*, vol. 86, pp. 1203–1221, June 1998.
[4] J. S. Montrym, D. R. Baum, D. L. Dignam, and C. J. Migdal, "InfiniteReality: A real-time graphics system," in *Proc. SIGGRAPH*, vol. 31, 1997, pp. 293–302.
[5] [Online]. Available: http://www.nvidia.com/
[6] S. Okamoto *et al.*, "A microprocessor with a 128-bit CPU, ten floating-point MACs, four floating-point divider, and an MPEG-2 decider," *IEEE J. Solid-State Circuits*, vol. 34, pp. 1698–1618, Nov. 1997.
[7] [Online]. Available: http://www.neomagic.com/
[8] Y. Aimoto *et al.*, "A 7.68-GIPS 3.84-GB/s 1-W parallel image-processing RAM integrating a 16-Mb DRAM and 128 processor," in *ISSCC Dig. Tech. Papers*, 1996, pp. 372–373.
[9] T. Watanabe *et al.*, "3-D CG media chip: An experimental single-chip architecture for three-dimensional computer graphics," *IEICE Trans. Electron.*, vol. EE77-C, pp. 1881–1887, Dec. 1994.
[10] ——, "A modular architecture for a 6.4-Gbytes/s 8-Mbit DRAM-integrated media chip," *IEEE J. Solid-State Circuits*, vol. 32, pp. 636–641, May 1997.
[11] K. Yoshda, T. Sakamoto, and T. Hase, "A 3-D graphic library for 32-bit microprocessor for embedded systems," *IEEE Trans. Consumer Electron.*, vol. 44, no. 3, pp. 1107–1114, Aug. 1998.
[12] D. Patterson *et al.*, "Intelligent RAM (IRAM): Chip that remember and compute," in *ISSCC Dig. Tech. Papers*, 1997, pp. 224–225.
[13] K. Murakami *et al.*, "Parallel-processing RAM chip with 256-Mb DRAM and quad processors," in *ISSCC Dig. Tech. Papers*, 1997, pp. 228–229.
[14] K. Inoue *et al.*, "A 10-Mb 3-D frame buffer memory with Z-compare and alpha blending," in *ISSCC Dig. Tech. Papers*, 1995, pp. 302–303.
[15] D. Landis, P. Hulina, S. Deno, L. Roth, and L. Coraor, "Evaluation of computing in memory architectures for digital image processing application," in *Int. Conf. Computer Design*, 1999, pp. 146–151.
[16] I. Koren, *Computer Arithmetic Algorithms*. Englewood Cliffs, NJ: Prentice-Hall, 1993.
[17] T. Sugibayashi *et al.*, "A 30-ns 256-Mb DRAM with multi-divided array structure," in *ISSCC Dig. Tech. Papers*, 1993, pp. 50–51.
[18] S. Y. Kung, *VLSI Array Processor*. Englewood Cliffs, NJ: Prentice Hall, 1988.

Yong-Ha Park received the B.S. degree in electrical engineering from Kyungpook National University, Taegu, Korea, and the M.S. degree in electrical engineering and computer sciences from the Korea Advanced Institute of Science and Technology (KAIST), Taejon, Korea, in 1996 and 1998, respectively. Since 1998, he has been working toward the Ph.D. degree at KAIST.

From 1998 to 2000, he was the Chief Researcher of RAMP-1 (RAM Processor) project. His current research interests are application-specific embedded memory logic focusing on single-chip 3-D graphics rendering.

Mr. Park received the Best Student Award for Outstanding Paper for the International Conference on VLSI and CAD in 1999. He is the Outstanding Design Award winner of the University LSI Design Contest at Asia South Pacific Design Automation Conference in 2001.

Seon-Ho Han was born in Kangwondo, Korea, in 1971. He received the B.S. and M.S. degrees in electrical engineering from Kangwon National University, Korea, in 1997 and 1999, respectively.

From 1998 to 1999, he was with Semiconductor System Laboratory (SSL), Korea Advanced Institute of Science and Technology (KAIST), Taejon, Korea, where he worked on DRAM, MMIC, and DLL. In 2000, he joined the Electrical Telecommunication and Research Institute (ETRI), Taejon, where he is engaged in frequency synthesizer and RF circuit design.

Jung-Hwan Lee received the Ph.D. degree in chemical engineering from Drexel University, Philadelphia, PA, in 1996, the M.S. degree from Korea Advanced Institute of Science and Technology, Taejon, Korea, in 1985, and the B.S. degree from Hanyang University, Seoul, Korea, in 1983.

In 1985, he joined Samsung Electronics Company, Ltd., Kyungkido, Korea, where he was involved in development of 64-kb, 256-kb, and 1-M SRAMs as a Process Integration Engineer. In 1989, he joined Hyundai Electronics Industries Company, Ltd., where he was engaged in development of 4-M, 16-M, and 64- M DRAMs as Process Integration Manager. He rejoined Hyundai Electronics in 1996 and led a device group for development of 0.13-μm DRAM technology. In 1997, he moved to the merged memory/logic (MML) device group and has managed the development of 0.35, 0.25, and 0.18-μm MML technology. His current interests are MML devices of beyond 0.15-μm and merged ferroelectric technology, as well as yield-up of MML products.

Hoi-Jun Yoo graduated from the Electronic Department of Seoul National University in 1983 and received the M.S. and Ph.D. degrees in electrical engineering from the Korea Advanced Institute of Science and Technology (KAIST), Seoul, in 1985 and 1988, respectively. His Ph.D. work concerned the fabrication process for GaAs vertical optoelectronic integrated circuits.

From 1988 to 1990, he was a Visiting Researcher at Bell Communications Research, Red Bank, NJ, and invented the two-dimensional phase-locked VCSEL array, the front-surface-emittting laser, and the high-speed lateral HBT. In 1991, he became Manager of a DRAM design group at Hyundai Electronics and designed a family of fast-1 M DRAMs and synchronous DRAMs including 256 M SDRAM. From 1995 to 1997, he was a Faculty Member of Kangwon National University. In 1998, he joined the faculty of the Department of Electrical Engineering at KAIST and currently leads a project team on RAMP (RAM Processor). In 2001, he founded a national research center, SIPAC (System Integration and IP Authoring Research Center), funded by the Korean government to promote worldwide IP authoring and its SOC applications. His current interests are SOC design, IP authoring, high-speed and low-power memory circuits and architectures, design of embedded memory logic, optoelectronic integrated circuits, and novel devices and circuits. He is the author of the books *DRAM Design* (in Korean, 1996) and *High Performance DRAM* (in Korean, 1999).

Dr. Yoo received the 1994 Electronic Industrial Association of Korea Award for his contribution to DRAM technology.