
휴대형 3D 그래픽 가속기를 위한 저전력/저면적 산술 연산기 회로 설계

김채현* · 신경욱**

A Design of Low-power/Small-area Arithmetic Units for Mobile 3D Graphic Accelerator

Chay-Hyeun Kim* · Kyung-Wook Shin**

이 논문은 2005년도 금오공과대학교 교내학술연구비 지원에 의한 연구결과의 일부임

요 약

본 논문은 휴대형 3D 그래픽 가속기를 위한 벡터 처리기, 누승기, 제산기 및 제곱근기 회로 설계에 관하여 기술한다. 설계된 연산기는 부동소수점 대신 OpenGL/ES에서 권장하는 16.16 고정소수점 방식을 사용하여 모바일 환경에서 저전력/저면적으로 동작하도록 하였다. 벡터 처리기는 RB 수체계 기반으로 설계되었으며 일반적인 4개의 승산기와 3개의 가산기로 구현한 방식에 비해 30%의 동작성능이 향상됐고, 10%의 면적 감소를 이루었다. 누승기, 제산기 및 제곱근기는 로그 수체계 기반으로 설계되었으며 이진수-로그 변환 시 룩업 테이블을 사용하지 않고 6-영역의 근사화 방법을 이용한 조합회로로 구현하였다. 누승기, 제산기 및 제곱근기는 일반적인 룩업 테이블로 구현한 방식과 비교하여 면적이 대폭 감소되었다.

ABSTRACT

This paper describes a design of low-power/small-area arithmetic circuits which are vector processing unit, powering unit, divider unit and square-root unit for mobile 3D graphic accelerator. To achieve area-efficient and low-power implementation that is an essential consideration for mobile environment, the fixed-point format of 16.16 is adopted instead of conventional floating-point format. The vector processing unit is designed using redundant binary(RB) arithmetic. As a result, it can operate 30% faster and obtained gate count reduction of 10%, compared to the conventional methods which consist of four multipliers and three adders. The powering unit, divider unit and square-root unit are based on logarithm number system. The binary-to-logarithm converter is designed using combinational logic based on six-region approximation method. So, the powering unit, divider unit and square-root unit reduce gate count when compared with lookup table implementation.

키워드

3D graphic accelerator, Arithmetic units, Redundant Binary, Logarithm number system

* Core Logic 연구원

접수일자 : 2006. 1. 16

** 금오공과대학교 전자공학과 교수

I. 서 론

2D 그래픽에서는 면에 국한된 표현만이 가능했으나 3D 그래픽에서는 면으로 이루어진 공간을 표현하기 때문에 그 표현범위가 대폭 넓어졌다. 3D 그래픽 처리는 특성 상 많은 양의 데이터와 수학적 연산이 필요하기 때문에 일반 범용 프로세서에서 전담하여 처리하기에는 한계가 있다. 고속의 범용 프로세서를 이용한 3D 그래픽 처리 방식이 사용되거나 그래픽 처리를 위한 하드웨어를 별도로 설계해야 한다.^[1,2]

휴대형 환경의 경우 멀티미디어 기능이 대폭 강화된 휴대폰과 그래픽 가속 칩을 탑재한 PDA의 등장 등 3D 그래픽이 구현될 수 있는 기술이 빠르게 발전하고 있다. 휴대형 3D 그래픽 기술은 과거 PC나 워크스테이션 급에서 3D 그래픽 기술이 발전한 것과 같은 유사한 과정을 거치게 될 것으로 전망되고 있다. 이러한 과정에서 가장 큰 문제점은 PC 수준의 고성능이면서 저전력 특성을 갖는 그래픽 가속 칩을 제작해야 한다는 것이다. 본 논문에서는 휴대형 환경에 맞게 OpenGL/ES에서 권장하는 16.16 고정 소수점 실수 표현방식[2]을 사용하여 저전력/저면적 벡터 처리기, 누승기, 제산기 및 제공근기를 설계하였다. 벡터 처리는 redundant binary(RB) 수체계를 사용함으로써, 가산트리단계에서 캐리전파의 지연을 없애 고속으로 동작할 수 있도록 하였다. 누승기는 기하단계의 조명모델에서 전반사나 소프트라이트 효과를 내기 위해 각의 코사인 값을 누승하는 연산을 담당한다. 일반적인 방법은 로그 수체계를 적용하여 AB 을 $2B \cdot \log_2(A)$ 로 계산하는 것이다. 기존의 누승기는 이진수-로그 변환 시 룩업 테이블을 이용하는 구조를 적용하고 있으나, 입력 비트수의 크기에 따라 테이블의 크기가 지수적으로 증가하므로 연산 정밀도를 허용오차 범위내로 줄이는 조합회로로 구현하여 면적이 대폭 감소되도록 설계하였다. 제산기 및 제공근기는 기하단계의 $1/W$ 의 계산 과정, 역의 전치행렬 계산과정, 벡터의 정규화(normalization) 과정에 필요한 연산으로써, 로그 수체계를 적용하여 이진수-로그, 로그-이진수 변환 시 조합회로로 구현하여 면적이 최소화 되도록 하였다.

본 논문은 다음과 같은 순서로 구성된다. II장에서는 3D 그래픽 렌더링 파이프라인에 사용되는 연산에 관해 기술하고, III장에서는 벡터 처리기 회로설계에 관해, IV장에서는 누승기, 제산기 및 제공근기의 회로설계에 관하여 기술한다. V장에서는 설계된 연산기의 설계검증과 성

능평가를 기술하며 마지막으로 VI장에서 결론을 맺는다.

II. 3D 그래픽 렌더링 파이프라인

렌더링 파이프라인의 주된 기능은 가상 카메라, 3차원 객체, 광원, 조명처리모델, 텍스처(texture) 등이 주어졌을 때 그것으로부터 2차원 이미지를 만들어내는 것이며, 응용 단계, 기하단계(geometry stage), 래스터화 단계(rasterization stage) 등으로 구성된다.^[2]

3D 그래픽 데이터 처리는 그 구조상 병렬성이 높고 많은 산술연산을 필요로 한다. 3D 그래픽 렌더링 파이프라인에서 필요한 산술연산은 기본적인 사칙 연산 외에 부가적인 연산들을 요구한다. 변환과정에서는 3D 그래픽 모델의 좌표를 변환하는 과정으로 4×4 행렬과 4×1 행렬의 곱셈연산이 주를 이루며 곱셈과 덧셈 연산으로 이루어진다. 라이팅(lighting) 과정은 3D 그래픽 모델의 각 정점에 빛 요소를 적용하여 해당 점점의 색을 구하는 과정으로 변환과정에서와 마찬가지로 곱셈과 덧셈은 음영처리 전 과정에서도 필요하다. 역 제공근은 빛 처리에서 계산된 벡터들을 단위벡터로 만들어 주는 과정에서 필요하게 되며 누승은 집중 조명광 계산과 전반사 성분 계산에서 필요하게 된다. 이외에 나눗셈 연산은 투영 변환 후 모델의 좌표 동차화 과정과 모델변환, 시야변환 행렬의 역의 전치행렬을 구하는 과정에서 필요하다. 표 1에 각 연산들의 3D 렌더링 파이프라인에서의 사용과정을 정리하였다.

표 1. 3D 그래픽 렌더링 파이프라인에서의 요구 연산
Table 1. Arithmetic types in 3D graphic rendering

연산 형태	사용 과정
가산, 감산, 승산	변환, 라이팅 전 과정
나눗셈	변환의 $1/W$ 과정 역의 전치 행렬 계산 과정
누승	라이팅의 집중 조명 처리 과정 라이팅의 전반사 처리 과정
역 제공근	라이팅의 벡터 정규화 과정

III. 벡터 처리기 회로 설계

본 장에서는 RB 수체계 기반의 벡터 처리기(vector

processing unit) 회로 설계에 관해 기술한다.

3.1 RB 기반의 벡터 처리기 회로 설계

그림 1은 설계된 벡터 처리기의 블록도이다. RB 수체계 기반의 2차원 승산기(Redundant Binary 2- Dimensional MULtiplier; RB2DMUL) 2개, RB 가산기 1개, RB-이진수 변환기 블록 등으로 구성된다. RB2DMUL은 $ax + by$ 연산을 한번에 수행하며 가산트리 단계에서 캐리전파가 없이 고속으로 동작할 수 있으며, 2개의 RB2DMUL에서 나온 중간값을 RB 가산기를 거쳐 최종 2의 보수체계로 변환하게 된다. RB 수체계에서 2의 보수 수체계로 변환 시에는 캐리 전파가 있으므로 하이브리드 구조를 적용하여 리플캐리 가산기(ripple carry adder)와 캐리선택 가산기(carry select adder)를 사용하여 구성하였다.^[3]

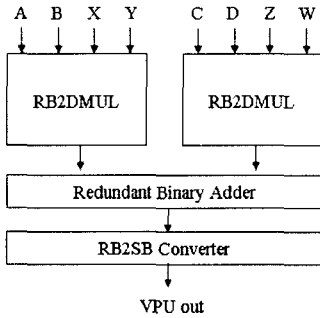


그림 1. 설계된 vector processing unit의 구조
Fig. 1 Architecture of vector processing unit

3.2 Redundant Binary 2-Dimensional Multiplier

설계된 RB2DMUL의 알고리즘은 다음과 같다. 2의 보수로 표현된 a, b, x, y 에서 중간값 T 는 다음 식(1)과 같이 표현된다. 식(1)을 RB 수체계로 변환하기 위하여 $\{-1, 0, 1\}$ 에 속하는 $\alpha_{i,j}$ 와 부호 반전 $\alpha^*_{i,j}$ 를 식(2)와 같이 인코딩한다. 예를 들면 $\alpha_{i,j} = 1$ 이면 $\alpha^*_{i,j} = -1$ 이다. RB 수체계에서 $ax + by$ 는 식(2)에 정의된 $\alpha_{i,j}, \alpha^*_{i,j}$ 를 식(1)에 적용하여 RB 부분곱 $P(i)$ 를 식(3)과 같이 정의하면 식(1)의 T 는 식(4)와 같이 N 개의 RB 부분곱의 합으로 단순화된다.

그림 2는 식(1) ~ 식(4)의 방법을 적용하여 설계된 RB2DMUL의 구조로써, 가산트리 단계에서 캐리전파 없이 $ax + by$ 연산을 한 번에 수행한다.

$$T = ax + by \tag{1}$$

$$= (u_{N-1,N-1} - \bar{v}_{N-1,N-1})2^{2N-2} + \sum_{i=0}^{N-2} \sum_{j=0}^{N-2} (u_{i,j} - \bar{v}_{i,j})2^{i+j} + 2^{N-1} \sum_{i=0}^{N-2} (\bar{u}_{i,N-1} - v_{i,N-1})2^i + 2^{N-1} \sum_{j=0}^{N-2} (\bar{u}_{N-1,j} - v_{N-1,j})2^j + 1$$

단, $u_{i,j} = a_{i,j}, v_{i,j} = b_{i,j}$ 은 비트 레벨 부분곱

$$\alpha_{i,j} \equiv u_{i,j} - \bar{v}_{i,j} \tag{2}$$

$$\alpha^*_{i,j} \equiv \bar{u}_{i,j} - v_{i,j}$$

$$P(i) = \alpha^*_{i,N-1}2^{N-1+i} + \sum_{j=0}^{N-2} \alpha_{i,j}2^{i+j} \tag{3}$$

$$(0 \leq i \leq N-2)$$

$$= \alpha_{N-1,N-1}2^{2N-2} + \sum_{j=0}^{N-2} \alpha^*_{N-1,j}2^{N-1+j}$$

$$(i = N-1)$$

$$T = \sum_{i=0}^{N-1} P(i) + 1 \tag{4}$$

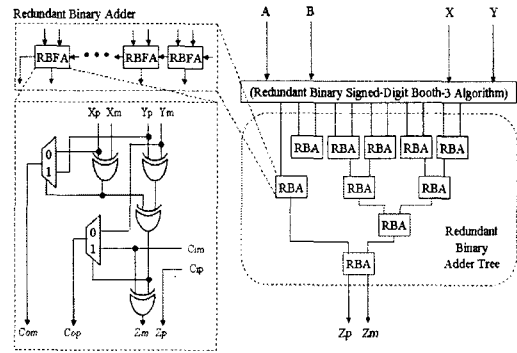


그림 2. Redundant binary 2-D 곱셈기
Fig. 2 Redundant binary 2-D multiplier

IV. 누승기, 제산기 및 제곱근기 회로 설계

4.1 LNS 기반의 누승기, 제산기 및 제곱근기 회로 설계

라이팅 연산에서 전반사나 스포트라이트를 계산하기 위한 누승기는 그림 3과 같이 구성된다. 설계된 누승기는

26-비트×12-비트를 승산하여 12-비트를 출력하는 절사 승산기가 사용되며, 가변 정밀도 승산기 생성기(Variable-Precision Multiplier Generator; VPM_Gen)^[4]를 사용하여 생성하였다. 로그에서 이진수의 변환 시 로그 역변환 테이블이 사용된다. 로그 역변환 테이블의 데이터는 Matlab을 이용하여 룩업테이블로 설계하였다. 테이블 크기는 입력 n-비트에 대하여 n*2n+2-비트의 저장 공간이 필요하며, 따라서 10-비트에 관하여, 5 KB의 저장 공간이 필요하다.

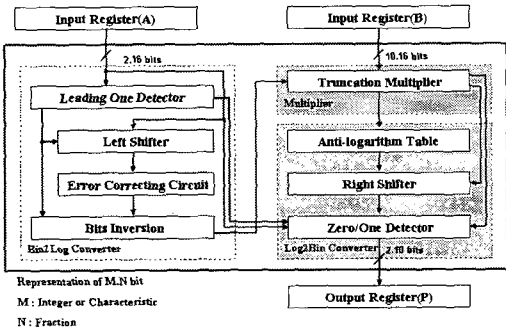


그림 3. 제안된 누승기의 구조
Fig. 3 Proposed powering circuit

기존의 제산 알고리즘은 크게 2가지 부류로 나뉜다. 첫째 방식은 가산/감산과 쉬프트 동작을 하는 방법이며, restoring 알고리즘, non-restoring 알고리즘, SRT 알고리즘 등이 이에 속한다. 둘째 방식은 승산을 이용하여 제산을 수행하는 방법으로, 분모·분자에 동일한 수를 곱하는 convergence 방식과 분모의 역수를 구해 곱하는 NR (Newton-Raphson) 방식 등이 이에 속한다. 본 논문에서는 입력을 로그수체계로 변환 후 각각 감산과 쉬프트를 사용하여 제산과 제곱근을 구현하였다. 그림 4와 그림 5는 설계된 제산기와 제곱근기의 전체 구조이며, 크게 나누어 이진수-로그 변환기, 감산기, 로그-이진수 변환기 등으로 구성된다. 설계된 제산기와 제곱근기는 이진수-로그 변환기와 로그-이진수 변환기의 구조는 동일하며, 로그값으로 변환된 수를 제산기는 감산하고 제곱근기는 1-비트 오른쪽 쉬프트 시킨다. 본 논문에서는 carry select 방식의 감산기를 사용하였으며, 1-bit 쉬프트는 로그값으로 변환된 입력 비트들의 배선으로 구현하였다. 로그값을 이진수로 변환하기 위해서는 이진수-로그 변환의 역변환을 거친다. characteristic에서 우선 leading one의 자리를 결정하고 fraction 부분을 leading one의 오른쪽에 위치시킨다.

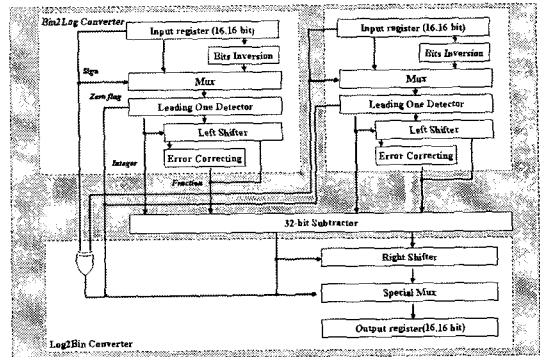


그림 4. 제안된 제산기 회로
Fig. 4 Proposed divider circuit

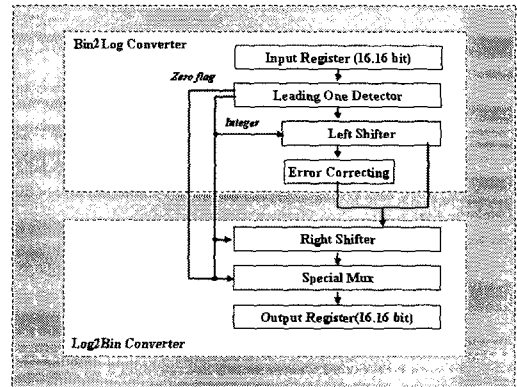


그림 5. 제안된 제곱근기 회로
Fig. 5 Proposed square root circuit

4.2 로그 변환기

일반적으로 로그수체계 이용 시 이진수-로그값의 변환은 룩업 테이블을 이용하였다. 이 방법은 필요한 정밀도까지 연산이 가능하나 그에 따른 테이블의 크기가 지수적으로 증가하여 정밀도를 허용오차 범위 내로 줄이는 조합회로로 구현하여 면적이 대폭 감소되도록 설계하였다.

Mitchell은 이진수 로그 분석에 대해 다음과 같이 정의하였다. 이진수 N 이 $2^j \leq N \leq 2^k$ ($j=0, \pm 1, \pm 2, \dots, k=0, \pm 1, \pm 2, \dots, k \geq j$)의 범위에 있을 때 $N = z_k \dots z_1 z_0 z_{-1} z_{-2} \dots z_j$ 와 같이 표현된다. N 은 식(5), (6)과 같이 Mitchell의 근사화 식으로 나타낸다.^[5]

$$N = \sum_{i=j}^k 2^i z_i = 2^k + \sum_{i=j}^{k-1} 2^i z_i = 2^k \left(1 + \sum_{i=j}^{k-1} 2^{i-k} z_i \right) = 2^k (1 + m)$$

(5)

$$N = 2^k(1 + m) \rightarrow \log_2 N = k + \log_2(1 + m) \quad (6)$$

m은 최상위 '1'의 오른쪽에 위치해 있는 모든 비트들을 포함하고, k는 로그의 특성(characteristic)으로 로그 수 체계로 변환된 수가 두개의 2의 거듭제곱 사이에 있는 것을 의미한다.

부동소수점 연산이나 로그 수 체계에서 사용되는 LOD(leading one detection)는 비트열들 중에 최상위 '1'의 위치를 찾는다. 본 논문은 병렬구조인, 최상위 '1'의 자리수를 1단계에 표현하는 방법을 사용하며 그림 6의 8-비트 LOD 회로는 상위 비트의 LOD 구현시 규칙적인 구조로 설계할 수 있다.

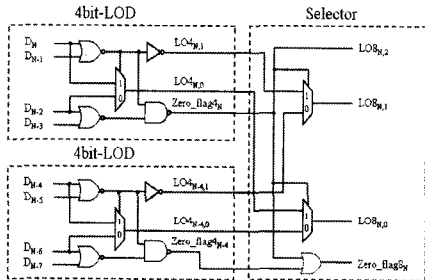


그림 6. 8-비트 LOD 회로도
Fig. 6 8-bit LOD circuit

N-비트만큼의 자리이동은 로그변환 시에 최상위 '1'의 오른쪽 위치에 소수부분을 제공하는 역할을 한다. 로그변환 시에는 왼쪽 쉬프트 기능만이 필요하므로 배럴 쉬프트를 이용하지 않고 구현하는 방법으로 설계하였다. 설계된 쉬프트는 배럴 쉬프트와 마찬가지로 입력 데이터 워드, 출력 데이터 워드, 제어 입력으로 구성되며 제어 입력의 비트 수 n에 따라 n개의 스테이지로 구성된다. 그림 7은 로그변환에 사용되는 왼쪽 16-비트 쉬프트이다.

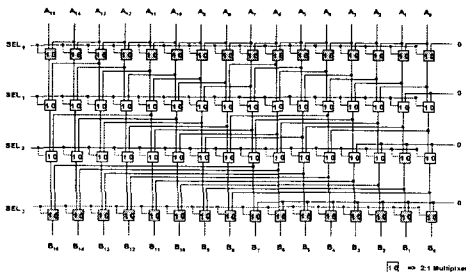


그림 7. 16-비트 left shifter
Fig. 7 16-bit left shifter

초기에 제안된 Mitchell의 방법은 소수 3번째 자리의 정밀도만 가지고 있으므로 정밀도를 높이기 위한 방법들이 Combet, Hall, Sangregory, Khalid 등에 의해서 제안되었다. 제안된 방법들 중 Khalid가 제안한 6-영역의 오차보상 방법^[6]은 소수 7번째 자리의 정밀도를 가지고 있다. 이를 이용하여 로그변환 시 에러를 최대한 줄일 수 있는 방법을 사용하였다. 식(7)은 Khalid의 6-영역의 오차 보상식이고, 그림 8은 식(7)의 회로 구현이다.

$$\text{fraction part} = \begin{cases} (m + (1/4)\bar{m}_{\text{offset}}) & \text{for } m \in [0, 0.0625) \\ (m + (1/4)\bar{m}_{\text{offset}} + 2^{-6}) & \text{for } m \in [0.0625, 0.25) \\ (m + 2^{-4} + 2^{-7} + 2^{-8}) & \text{for } m \in [0.25, 0.375) \\ (m + 2^{-4} + 2^{-6} + 2^{-7}) & \text{for } m \in [0.375, 0.625) \\ (m + 2^{-4} + 2^{-7}) & \text{for } m \in [0.625, 0.75) \\ (m + (1/4)\bar{m}_{\text{offset}}) & \text{for } m \in [0.75, 1.0) \text{ at } \bar{m} = (1 - m_{\text{offset}} - 2^{-4}) \end{cases} \quad (7)$$

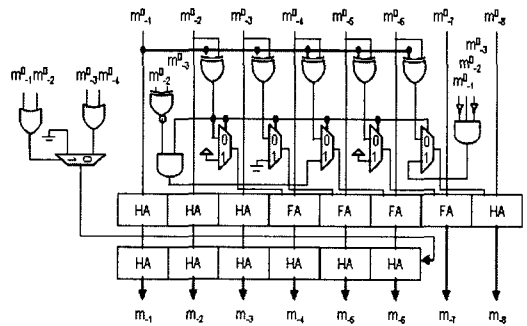


그림 8. 오차 보상 회로
Fig. 8 Error compensation circuit

V. 설계 검증 및 성능 평가

설계된 산술 연산기들은 Verilog-HDL로 모델링되었으며 기능검증은 Matlab 모델링으로부터 50,000개의 랜덤 테스트 벡터를 생성하고, 이를 Verilog testbench에서 읽어 ModelSim에서 논리 시뮬레이션을 수행하였다. 하드웨어 면적 및 타이밍 정보 추출을 위한 논리회로 합성은 0.25-μm CMOS 셀 라이브러리와 Synopsys를 이용하여 회로를 합성하였다. 전력소모 측정은 Matlab에서 생성된 50,000개의 테스트 벡터를 VCS(Verilog Compiled Simulator)를 사용하여 연산기 내부의 switching activity를 산출한 후 Power Compiler를 통해 평가하였다. 또한 Matlab에서 teapot 예제를 통하여 설계된 연산기들의 동작을 검증하였다.

5.1 벡터처리의 설계 검증 및 성능 분석

설계된 벡터 처리기의 검증은 Matlab 모델링으로부터 8개의 입력 a, x, b, y, c, z, d, w 각각의 50,000개의 랜덤 테스트 벡터를 생성하였고, 이를 Verilog testbench에서 읽어 논리 시뮬레이션을 수행한 결과와 Matlab 모델링으로 생성된 결과를 비교하여 논리 동작을 검증하였다. 설계된 연산기들의 합성결과에 의하면 RB2DMUL 블록은 약 16,000 게이트, RB 가산기 블록은 약 1,200 게이트, RB-SB 컨버터 블록은 약 800 게이트로 구현되었다. 설계된 벡터 처리기는 일반적인 승산기 4개와 가산기 3개로 구현한 것과 비교하여 30%의 동작 성능을 향상시켰고 10%의 면적 감소를 이루었다. 그림 9는 teapot의 원본 이미지이며, 그림 10은 그림 9의 이미지에서 크기를 절반으로 감소하고 x축으로 90° 회전하였으며 x축으로 3, y축으로 3만큼 자리이동 한 모델 변환의 결과이다. 표 2는 벡터처리의 성능 분석 결과이다.

표 2. 벡터 처리기의 성능 분석
Table 2. Performance of vector processing unit

	Conventional	Proposed
게이트 수	39,000 (1.0)	35,500 (0.9)
동작 주파수	55-Mhz@2.5V	85-Mhz@2.5V
전력 소모 특성	13.61-mW	14.27-mW

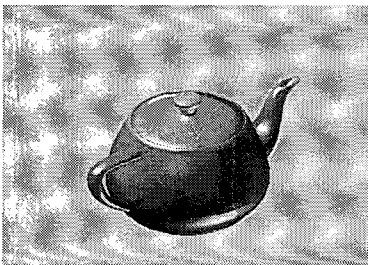


그림 9. teapot 원본 이미지
Fig. 9 Original image of teapot

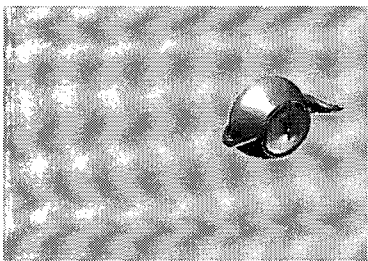


그림 10. 모델 변환 테스트 결과
Fig. 10 Image of model transformation

5.2 누승기, 제산기 및 제곱근기의 설계 검증 및 성능 분석

누승기, 제산기 및 제곱근기는 Matlab에서 생성된 테스트 벡터 50,000개와 ModelSim으로 기능 검증하였으며 식(8), (9)와 같이 연산 정밀도를 분석하였다. PM은 Matlab에서 생성된 결과값, PV는 Verilog 모델에서 출력된 결과값, M은 테스트 벡터 개수를 나타낸다. 평균 퍼센트 오차는 정확한 결과 값에 비해 오차가 차지하는 비중의 척도를 나타낸다. 연산기들의 합성결과 컨버터는 약 300 게이트로 구현되어 일반적인 룩업 테이블 크기(룩업 테이블로 구현 시 입력 n-비트에 대하여 n*2ⁿ⁺²-비트의 저장 공간이 필요)에 비하여 게이트 수가 대폭 감소되었다. 설계된 누승기는 일반적인 룩업 테이블을 사용하여 부동소수점으로 구현한 방식과 비교하여 20%의 면적으로 구현됐으며, 평균 오차가 0.002로 소수점 이하 9-비트내의 연산 정밀도를 갖는 것으로 평가 되었으며 평균 퍼센트 오차는 1.6%인 것으로 평가되었다. 설계된 제산기 및 제곱근기는 기존의 룩업 테이블을 사용하여 구현한 방식과 비교하여 면적이 대폭 감소되었으며, 연산 정밀도 분석결과 제산기와 제곱근기의 평균에러는 각각 736, 5.4인 것으로 평균 퍼센트 오차는 3.8%, 4.2% 인 것으로 평가되었다. 누승기, 제산기 및 제곱근기의 전력소모 특성은 VCS와 Power Compiler를 통해 평가하였으며 표 3에 성능 분석을 정리하였다. 누승기의 동작을 검증하기 위해 그림 11의 원본 이미지와 그림 12의 테스트 결과를 비교한 결과 이미지의 차이가 거의 없는 것으로 평가되었다.

$$E_v = \frac{\sum |PM - PV|}{M} \tag{8}$$

$$PE_v = \frac{\sum \left| \frac{PM - PV}{PM} \right| \times 100}{M} \tag{9}$$

표 3. 누승기, 제산기 및 제곱근기의 성능 분석
Table 3. Performance of powering, divider and square root circuits

		누승기	제산기	제곱근기
게이트 수		6,350	3,130	1,280
동작 주파수		50-Mhz@2.5V	50-Mhz@2.5V	60-Mhz@2.5V
전력 소모 특성		1.28-mW	1.53-mW	1.35-mW
연산 정밀도 분석	평균오차	0.002	736	5.4
	평균 퍼센트 오차	1.6%	3.8%	4.2%

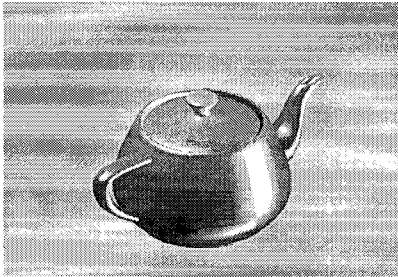


그림 11. Teapot 원본 이미지
Fig. 11 Original image of teapot

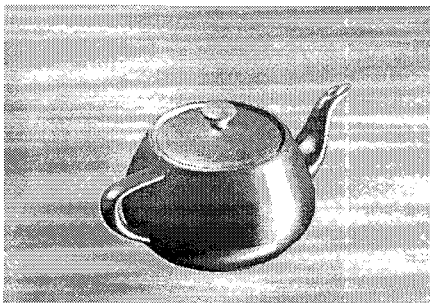


그림 12. 라이팅 연산의 테스트 결과
Fig. 12 Image of lighting operation

※ 반도체설계교육센터(IDECE)의 CAD Tool 지원에 감사드립니다.

참고문헌

- [1] 이범렬, 류성원, 이은주, 박재형, “모바일 3D API 기술 표준화 연구”, 전자통신동향분석, vol. 20, no. 4, Aug. 2005.
- [2] E. Tomas, “REAL-TIME RENDERING 제 2판”, 정보문화사, Nov. 2003.
- [3] “Using Fixed-Point Instead of Floating Point for Better 3D Performance”, <http://www.devx.com/Intel/Article/16478>
- [4] Kyung-Wook Shin, Bang-sup Song, Kantil Bacrania, “A 200-Mhz Complex Number Multiplier Using Redundant Binary Arithmetic”, IEEE Solid-state Circuits, vol. 33, no. 6, pp. 904-909, June. 1998.
- [5] J. W. Lee, S. K. Hwang, K. W. Shin, “저전력 DSP 응용을 위한 오차보상을 갖는 가변 정밀도 승산기 코어 생성기”, 한국통신학회논문지, vol. 30, pp. 129-136, Dec. 2004.
- [6] J. N. Mitchell, “Computer Multiplication and Division Using Binary Logarithms”, IRE Trans. electronic Computer, vol. 11, pp. 512-517, Aug. 1962.
- [7] H. Suzuki, H. Morinaka, H. Makino, Y. Nakase, K. Mashiko, T. Sumi, “Leading-Zero Anticipatory Logic for High-Speed Floating Point Addition”, IEEE J. Solid-state Circuits, pp. 1157-1164, 1996.
- [8] H. A. Khalid, E. S. Raymond, “CMOS VLSI Implementation of a Low-Power Logarithmic Converter”, IEEE Trans. on Computer vol. 52, no. 11, pp. 1421-1433, Nov. 2003.

VI. 결 론

본 논문에서는 3D 그래픽 가속기에 쓰이는 산술 연산기를 설계하였다. 설계된 연산기들은 벡터 처리기, 누승기, 제산기 및 제곱근기로서 OpenGL/ES에서 권장하는 16.16 고정소수점 방식으로 설계하였다.

벡터 처리기는 RB 수체계를 적용하여 승산시 가산단계의 캐리전파가 없어 승산기 4개와 가산기 3개로 구현한 방식보다 30%의 동작 성능이 향상됐다.

누승기, 제산기 및 제곱근기는 로그 수체계 기반으로 설계하여 이진수-로그 변환시 룩업 테이블을 사용하지 않고 조합회로로 구현함으로써 면적이 최소화 되도록 설계하였다. 로그 수체계 기반으로 설계된 누승기 및 제곱근기는 승산, 제산 등의 연산을 가산, 감산 등으로 구현할 수 있으므로 이후 역수, 제곱근의 역수 등의 특수 연산(special function) 등의 설계에도 적용할 수 있을 것으로 예상된다.

본 논문에서 설계된 연산기들은 모바일 환경의 3D 그래픽 가속기 및 디지털 신호 처리 등에 활용될 수 있을 것으로 판단된다.

저자소개



김 채 현(Chay-Hyeun Kim)

2004년 2월 금오공과대학교
전자공학과 졸업

2006년 2월 금오공과대학교
전자공학과 공학석사

2006년 3월 ~ 현재 코아로직 연구원

※관심분야: SoC 설계, 3D 그래픽 가속기용 집적회로
설계, 반도체 IP 설계



신 경 옥(Kyung-Wook Shin)

1984년 2월 한국항공대학교 전자공
학과 졸업

1986년 2월 연세대학교 대학원 전자
공학과(공학석사)

1990년 8월 연세대학교 대학원 전자공학과(공학박사)

1990년 9월 ~ 1991년 6월 한국전자통신연구소 반도체
연구단(선임연구원)

1991년 7월 ~ 현재 금오공과대학교 전자공학부(교수)

1995년 8월 ~ 1996년 7월 University of Illinois at Urbana-
Champaign(방문교수)

2003년 1월 ~ 2004년 1월 University of California at San
Diego(방문교수)

※관심분야: 통신 및 신호처리용 SoC 설계, 정보보호
SoC 설계, 반도체 IP 설계