
3차원 모바일 그래픽 응용을 위한 스캔 변환 프로세서의 설계

최병윤* · 하창수* · Zoran Salcic**

Design of Scan Conversion Processor for 3-Dimensional Mobile Graphics Application

Byeong-yoon Choi* · Chang-soo Ha* · Zoran Salcic**

이 논문은 동의대학교 연구년 과제와 정통부 선도 기반 과제의 지원에 의한 결과임

요 약

본 논문에서는 3개의 정점으로 표현된 삼각형을 화면 좌표, 깊이 좌표, 색 등의 픽셀 정보로 바꾸는 스캔 변환 프로세서를 설계하였다. 설계된 스캔 변환 회로는 삼각형 변을 따라 스캔 라인을 생성한 후, 스캔 라인을 픽셀로 변환하는 스캔 라인 방식을 사용하였다. 그리고 좌변-상변 픽셀 채움 규칙을 지원함에 의해서 인접한 삼각형의 공유 변에 대한 픽셀이 중복 처리와 미처리 되는 것을 방지한다. 스캔 변환 프로세서는 약 21,400개의 게이트로 구성되며, 0.35 μ m CMOS 공정 조건에서 약 80 Mhz의 동작 주파수를 가진다. 스캔 변환 프로세서는 최대 80 Mpixels/sec의 픽셀 생성율을 갖고 있어 3차원 모바일 그래픽 분야에 적용 가능하다.

ABSTRACT

In this paper, the scan conversion processor which converts the triangle represented by three vertices into pixel-level screen coordinates, depth coordinate, and color data is designed. The processor adopts scan-line algorithm which decomposes triangle into horizontal spans and then transforms the span into pixel data. By supporting top-left filling convention, it ensures that triangles that share an edge do not produce any dropouts or overlaps between adjacent polygons. It consists of about 21,400 gates and its maximum operating frequency is about 80 Mhz under 0.35 μ m CMOS technology. Because its maximum pixel rate is about 80 Mpixels/sec, it can be applicable to mobile graphics application.

키워드

스캔 변환 회로, 래스터라이저, 3차원 그래픽스, 모바일 디바이스, 픽셀 채움 규칙, SoC

I. 서 론

최근 3차원 그래픽 분야는 3차원 게임, 의료 영상, 항공 시뮬레이션과 모바일 분야 등에 응용이 확대되고 있다. 특히 멀티미디어 통신 서비스와 반도체 기술의 발달로 최근 모바일 기기가 단순한 통신 혹은 컴퓨터 기기 수

준을 뛰어 넘어 종합적인 엔터테인먼트 장비가 되어 감에 따라 고수준의 그래픽 처리를 필요로 하고 있다[1]. 특히 3차원 컴퓨터 그래픽 분야는 처리해야 할 연산 양이 많으므로 실시간 처리를 하려면 호스트 CPU 외에 별도의 하드웨어 그래픽 가속기가 필요하다. 3차원 그래픽스 분야에서 3차원 장면을 구성하기 위한 기본적인 요

* 동의대학교 컴퓨터공학과

** University of Auckland University, New Zealand

소를 프리미티브(primitive)라 하는데, 프리미티브의 종류는 점(point), 선(line), 그리고 폴리곤(polygon)으로 구성된다. 현재 복잡한 폴리곤은 하드웨어 및 소프트웨어 구현 용이성으로 삼각형의 집합으로 분할되어 구현된다. 폴리곤 기반의 3차원 그래픽 렌더링 과정은 크게 기하학 처리(geometry processing)와 렌더링(rendering) 부분으로 나눌 수 있다. 기하학 처리 과정은 객체에 대한 좌표 변환과 빛에 대한 계산이 이루어지는 과정으로서 부동 소수점 연산을 주로 사용한다. 기하학 처리 과정을 거치면서 각 정점(vertex)들의 정보는 좌표 및 깊이 정보(xyz), 컬러 정보(RGBA) 등을 생성하게 된다[2]. 렌더링 단계는 기하학 처리가 이루어진 후 변환된 프리미티브들을 픽셀 값(pixel value)으로 바꾸어 프레임 버퍼(frame buffer)안에 저장하는 과정이다.

본 논문에서는 삼각형을 나타내는 3개의 정점(vertex) 정보를 받아서, 삼각형 내부에 속하는 픽셀 좌표를 찾아내고, 해당 픽셀에서의 컬러 정보 등의 속성을 고속으로 생성하는 스캔 변환 회로를 설계하였다. 스캔 변환을 위해 사용하는 방식은 크게 스캔 라인(scan-line) 방식[3-9]와 3개의 에지 함수를 사용하는 타일 트래버설(tile traversal) 방식[10-11]로 나뉜다. 타일 트래버설 방식은 병렬 처리가 용이하지만 하드웨어가 복잡하고 픽셀 채움 기법 적용이 어렵다. 따라서 본 논문의 스캔 변환 회로는 알고리즘이 간단하고 하드웨어 구현이 용이한 스캔 라인 방식을 사용하였다. 그리고 3차원 그래픽 모델링의 경우 하나의 장면이 여러 개의 삼각형으로 분해되는데, 개별 삼각형 처리 경우 에지(edge)를 공유하는 인접 삼각형을 고려해야 한다. 즉, 공유된 에지에 놓이는 픽셀이 개별 삼각형 처리 방식에 따라 중복 처리 혹은 미처리(dropout)를 야기 할 수 있으므로, 이를 방지하기 위해 픽셀 채움 규칙을 만족해야 한다. 기존 스캔 변환 회로의 경우 픽셀 채움 기능을 내장하고 있지 않아 이러한 문제가 존재한다. 본 논문에서 설계한 스캔 변환 회로는 스캔 라인 스캔 변환 방식과 Direct3D와 OpenGL API의 좌변-상변 채움 규칙(top-left filling convention) [12-13]을 지원하도록 설계하였다.

본 논문의 구성은 다음과 같다. 제 2장에서는 스캔 라인 방식의 스캔 변환 동작과 픽셀 채움 규칙을 살펴보고, 제 3 장에서는 모바일 환경에 적합한 스캔 변환 프로세서의 성능 조건과 하드웨어 설계를 기술하였다. 제 4

장에서는 설계한 회로에 대한 검증과 성능 분석을 하였으며, 마지막으로 결론을 기술하였다.

II. 스캔 라인 방식의 스캔 변환 동작과 픽셀 채움 규칙

2.1 스캔 라인 스캔 변환 동작

3차원 그래픽 시스템은 그림 1과 같은 구조를 가지며, 스캔 변환 회로의 경우 렌더링 프로세서 내에 위치하며, 기하 처리 프로세서와 인터페이스 된다. 그림 1에서 삼각형 셋업(triangle setup) 모듈은 스캔 변환 동작에서 필요한 정점 간의 기울기 등의 파라미터를 계산하는 모듈이다[14]. 스캔 변환 동작은 기하학 처리 과정을 거쳐 변환된 좌표 값과 컬러 값 등의 정보로 구성된 세 개의 정점(vertex)을 가지고, 그림 2와 같이 삼각형 내부를 픽셀 단위로 구분하여 삼각형 내부에 대한 픽셀 정보(X, Y, Z, 컬러 정보 등)를 계산하는 과정이다. 삼각형의 세 점 V_0, V_1, V_2 가 들어오면 y축 값을 기준으로 정렬한다. 같은 y 좌표에 대하여 삼각형 변의 시작과 끝 x 좌표를 연결하는 수평한 선을 스패ن(span)이라 부른다. 스캔 라인 방식의 스캔 변환은 세 정점을 기준으로 하는 삼각형을 입력 받아서 y 축을 기준으로 하여 삼각형을 스패스로 분해한 후, 각각의 스패스를 x 축 방향으로 스캔하면서 스패스를 픽셀의 집합으로 분해하는 방식이다. x축 방향 스캔 과정에 각 픽셀 성분의 x축 방향 기울기 값을 더하여 해당 좌표에 대응하는 픽셀 정보를 구하게 된다.

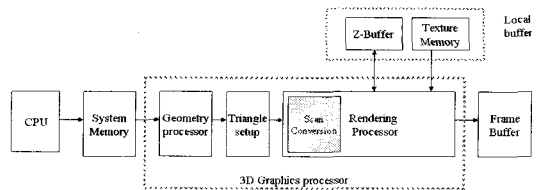


그림 1. 일반적인 3차원 그래픽 시스템 구조
Fig. 1. Typical 3D graphics system architecture

이러한 스캔 라인 동작을 하드웨어로 구현하려면 삼각형의 에지를 따라 가면서 스패스의 시작과 끝 위치를 계산하기 위한 2개의 보간기(interpolator) 회로와 함께 픽셀 속성, 즉 좌표 값(x, y, z)과 색(color) 등에 대한 보간기

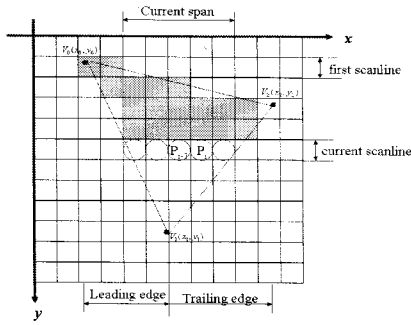


그림 2. 스캔 라인 스캔변환 방식
Fig. 2. scan-line scan conversion

회로가 필요하다. 그림 2의 P_i 지점에 대한 보간 (interpolation) 동작은 식 (1)과 같다. 단, 이러한 보간 동작에서 X 좌표는 1만큼 증가한다.

$$\begin{aligned}
 P_i : z_i &= z_{i-1} + \frac{\partial z}{\partial x} & (1) \\
 r_i &= r_{i-1} + \frac{\partial r}{\partial x} \\
 g_i &= g_{i-1} + \frac{\partial g}{\partial x} \\
 b_i &= b_{i-1} + \frac{\partial b}{\partial x}
 \end{aligned}$$

보간 동작에 사용되는 증분 값은 삼각형 셋업과정에 서 결정되어 보간 과정에 사용된다. 평면 특성상 픽셀 속 성의 증분 값은 삼각형 내에서 일정하므로 한번만 계산 하면 된다.

2.2 좌변-상변 픽셀 채움 규칙

삼각형 에지(edge)가 픽셀 중심(center)을 지나가는 경 우 해당 픽셀을 렌더링 할지 판단이 필요하다. 사각형을 그림 3과 같이 2개의 삼각형으로 나누어 처리할 경우 2 개의 삼각형에 공유하는 변이 픽셀 중심을 지나는 경우 픽셀 채움 규칙에 따라 공유 에지 상에 존재하는 픽셀을 2번 중복해서 렌더링하거나, 상황에 따라 두 개의 삼각 형이 해당 픽셀을 렌더링하지 않는 상황(dropout)이 발 생할 수 있다. 따라서 이러한 문제를 해결하기 위해 OpenGL과 Direct3D 그래픽 라이브러리의 경우 좌변-상 변 채움 규칙(top-left filling convention)을 채택하고 있 다. 상변 또는 좌변이 픽셀 중심을 지나갈 경우 해당 픽 셀을 렌더링 한다. 즉 폴리곤 내에 속하는 것으로 판단한 다, 반면 하변과 우변이 픽셀 중심을 지나갈 경우는 렌더

링하지 않는다. 즉 폴리곤 외부에 존재하는 것으로 판단 한다. 단, 픽셀 중심이 삼각형 에지와 교차하지 않고 완 전히 포함되는 경우는 모두 렌더링을 수행한다. 그림 3 에서 점(dot)으로 채워진 픽셀은 렌더링 되는 픽셀을 나 타내며, 점이 없는 부분은 삼각형 외부로 처리되는 픽셀 을 나타낸다.

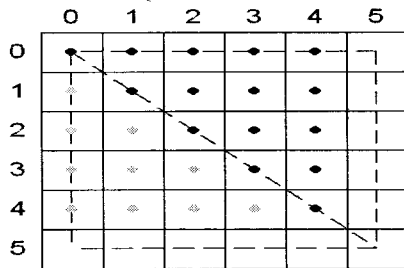


그림 3. 삼각형에 대한 좌변-상변 채움 규칙
Fig. 3. Top-left filling rule for triangle

좌변-상변 채움 규칙을 기존 스캔-변환 동작에 포함 시키려면, 삼각형의 정점 y 좌표와 스패의 시작과 끝 지 점의 x 좌표에 대한 두 가지 조정 처리가 필요하다.

첫째, 스캔 변환 동작이 시작되는 가장 낮은 y의 좌표 와 가장 큰 y값에 식(2)의 천장(ceiling) 연산이 필요하다. 여기서 y_0 와 y_1 은 삼각형의 가장 작은 y와 가장 큰 y 좌 표를 나타낸다.

$$\begin{aligned}
 y_{min} &= ceiling(y_0) = \lceil y_0 \rceil & (2) \\
 y_{max} &= ciling(y_1) - 1 = \lceil y_1 \rceil - 1
 \end{aligned}$$

여기서 $\lceil m \rceil$ 는 천장 함수(ceiling function)로 m보 다 작지 않은 가장 작은 정수를 생성한다.

둘째, 스패(span)의 시작 과 마지막 x 좌표 값도 천장 함수를 통해 정수 값으로 조정된다. x_{min} 과 y_{max} 는 현재 스패의 가장 작은 x 좌표와 가장 큰 x 좌표를 나타낸다.

$$\begin{aligned}
 x_{init} &= ceiling(x_{min}) = \lceil x_{min} \rceil & (3) \\
 x_{end} &= ceiling(x_{max}) = \lceil x_{max} \rceil - 1
 \end{aligned}$$

x_{init} 와 x_{end} 는 조정된 현재 스패의 시작과 끝 지점 의 정수 x 좌표를 나타낸다.

III. 스캔 변환 프로세서의 하드웨어 설계

본 장에서는 스캔 라인 프로세서의 성능 요구 조건, 스캔 라인 스캔 변환 동작의 하드웨어 구현 기법과 하드웨어 설계를 기술한다.

3.1. 성능 조건

3차원 그래픽 파이프라인에서 최고의 성능을 만들려고 하면 내부 파이프라인 단계에서 병목 현상이 발생하지 않아야 한다. 기하학 처리 단계에서는 그래픽 프리미티브가 객체 좌표(x, y, z)에서 화면 좌표(X_{screen}, Y_{screen})로 바뀌므로 각 정점 당 복잡한 변환 동작이 필요하다. 참고 문헌 [2, 6]에 따르면 일반적으로 하나의 영상이 평균 10,000개의 삼각형으로 구성되며, 각 삼각형은 100 개의 픽셀로 이루어진다. 이러한 조건에서 1 초에 적어도 30 개의 영상프레임이 발생되고, 각각의 삼각형이 3개의 정점을 갖는다고 가정하면, 실시간 동작 특성을 얻으려면 약 36.9 MFLOPS(Million Floating Point Operations Per Second)의 부동 소수점 연산 성능이 필요하다[6]. 이러한 필요조건은 현재 전용 그래픽 프로세서로 만족 가능하다. 따라서 기하학 처리 단계는 더 이상 그래픽 시스템에서 성능을 제약하는 병목 부분이 아니라고 판단할 수 있다. 반면 픽셀 처리 단계에서는 위에서 설정한 그래픽 영상 조건에 의해 영상 프레임 당 약 백만 개, 즉 100 pixels/triangle × 10,000triangles 의 픽셀이 처리되어야 한다. 즉 1 초에 30개의 영상이 처리될 경우 3 천만 개 이상의 픽셀이 계산되어야 함을 의미한다. 따라서 이러한 실시간 조건을 만족하려면, 스캔 변환 회로가 하나의 픽셀을 생성하는데 필요한 요구 시간은 식(4)과 같이 33.3 ns보다 작아야 한다. 즉 픽셀 생성율은 30 Mpixels/sec이다.

$$\begin{aligned} \text{scan conversion time} & \quad (4) \\ & \leq \frac{1}{\text{number of pixels per second}} \\ & \leq \frac{1}{30 \times 10^6} \end{aligned}$$

그러나 현재 3차원 그래픽 분야의 특성은 세밀한 모델링의 특성으로 영상 당 폴리곤의 수는 급격히 증가하며, 상대적으로 폴리곤당 픽셀의 수는 감소하고 있는 추세이다. 따라서 본 연구에서는 모바일 응용과 식 (4)을

고려하여 50 Mpixels/sec을 스캔 변환 회로의 목표 성능으로 설정하였다.

3.2. 스캔 라인 방식의 구현 기법

설계한 스캔 변환 프로세서는 3개의 정점들을 받아들여서, 정점의 x,y값을 비교하여 삼각형을 그림 4와 같이 4가지 유형으로 구분하여 처리하는 방식을 채택하였다[5, 7].

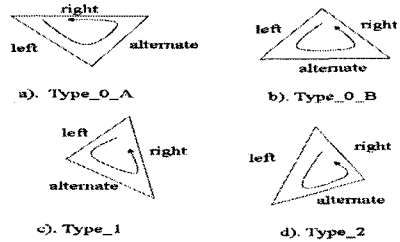


그림 4. 삼각형의 형식
Fig. 4. Types of triangle

하드웨어 구현이 용이하도록 Type_1 과 Type_2 형식은 스캔 변환 회로 내부적으로 2개의 삼각형(Type_0_A 와 Type_0_B 형식)으로 나뉘어 처리된다. 그림 5는 Type_2 삼각형 형식이 2개의 삼각형으로 분할되는 방식을 나타낸다. 그림 5와 같이 분할할 경우 스캔 라인 변환 방식에 픽셀 채움 규칙을 통합하는 것이 용이하다.

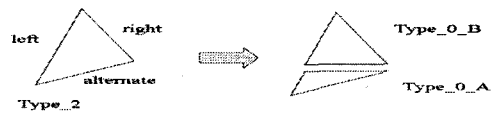


그림 5. Type-2 삼각형의 2개의 삼각형으로 분할
Fig. 5. Type-2 triangle decomposition into two triangles

스캔 라인 스캔 변환 동작은 크게 3 단계로 나뉜다. 삼각형 셋업(triangle setup) 단계: 삼각형을 스캔 변환하는데 필요한 모든 파라미터 값(삼각형의 각 변의 기울기, 보간해야 할 픽셀 속성(색깔, 깊이 등)성분에 대한 x, y 방향과 에지 방향의 기울기를 계산하는 과정으로, 이 과정에서 계산되는 정보는 폴리곤에 대해 한번만 계산하면 되기 때문에 폴리곤 처리 단계라고도 한다.

변 처리(edge processing): 삼각형 셋업 단계에서 계산된 기울기 값을 사용하여, 삼각형을 스캔 단위로 분할한다.

스팬 처리(span processing) : 변 처리 과정에서 생성된 스패에 대해서, 실제 각 픽셀에 대응하는 x, y 좌표 값과 속성 값을 구하는 단계로 스패의 시작 부분의 속성 값에 픽셀 속성의 x 방향 기울기를 순차적으로 더함에 의해 구현된다.

이러한 스캔 라인 스캔 변환 방식은 각 단계가 독립적으로 분리 가능하므로 파이프라인 방식으로 처리될 수 있다. 본 연구에서는 적은 면적 조건을 만족시키기 위해, 비 파이프라인 방식으로 스캔 변환 회로를 구현하였다. 그림 2의 스캔 라인 스캔 변환 방식에서 픽셀 채움 규칙을 적용하기 위해, 식(3)과 식(4)을 각 스패와 정점에 적용해야 한다. 픽셀 채움 규칙을 만족시키기 위해 고정 소수점 형태의 값을 갖는 스패의 시작과 끝 위치에 천장 함수를 적용하여 정수 값의 x, y 좌표를 생성한다. 스패의 x 좌표 정보와 천장 함수를 적용한 결과 좌표 사이의 관계를 그림 6(a)에 나타내었다. 픽셀 채움 규칙을 만족하는 정수 값 형태의 x_{init} 와 x_{end} 는 조정된 스패의 시작 위치와 종료 위치를 나타낸다. 스패 처리하는 동안 카운터-x(count_x) 레지스터는 x_{init} 로 초기화 된 후, 1씩 증가하면서, 카운터-x 레지스터가 x_{end} 값과 같게 되면 스패 처리를 종료하게 된다. 이 과정에 다른 픽셀 성분은 식(1)과 같이 x 방향의 증분을 순차적으로 더해서 해당 픽셀 위치의 픽셀 속성을 생성한다.

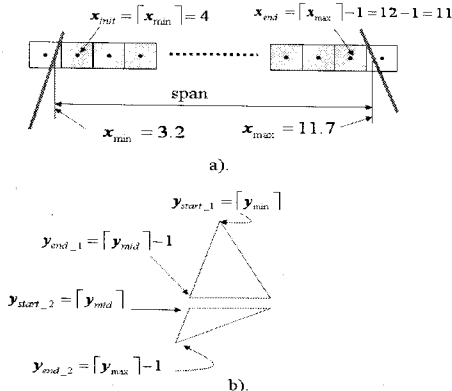


그림 6. 픽셀 채움 규칙의 구현
(a). 좌변과 우변의 픽셀 채움 동작
(b). 상변과 하변의 픽셀 채움 동작
Fig. 6. Implementation of Pixel filling rule
(a). pixel filling for left and right edge
(b). pixel filling for top and bottom edge

그림 6(b)은 Type 2의 삼각형이 2개의 삼각형으로 나뉘어 처리되는 경우, 변 처리의 시작과 종료 위치를 결정하는 상변과 하변에 대한 픽셀 채움 동작을 지시한다. 카운터-y 레지스터(count-y)는 초기에 y_{start} 로 초기화된 후, 각 변 처리 동작 후에 1씩 증가 한다. 카운터-y 레지스터 값이 y_{end} 와 같아지는 경우, 하나의 삼각형 처리가 종료 되게 된다. 상변과 하변에 대한 채움 처리를 위한 y_{start} 와 y_{end} 계산 동작은 폴리곤당 한번만 계산하면 되므로, 초기화 과정에 처리되도록 하였다. 천장 함수(ceiling)와 천장 함수 결과에서 1을 빼는 동작을 구현하기 위해 그림 7과 같은 천장 함수 구현 회로를 설계하였다. 입력 정보가 고정 소수점 형식의 실수 값 형식(M.N)을 가진다. 여기서 M, N은 각각 정수부와 실수부의 비트 길이를 나타낸다.

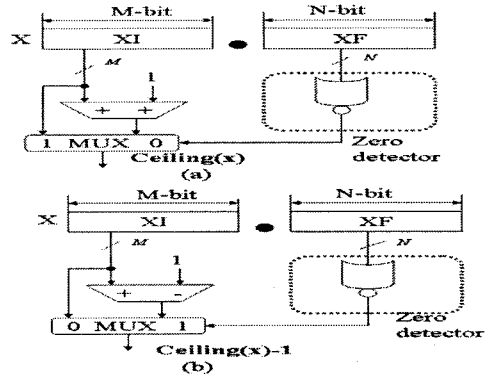


그림 7. 천장 함수 구현 방안
(a). ceiling(x) 함수
(b). ceiling(x)-1 함수

Fig. 7. Implementation of ceiling function
(a). ceiling(x) function
(b). ceiling(x)-1 function

3.3. 스캔 변환 프로세서의 하드웨어 설계

스캔 변환 프로세서는 입력 레지스터, 내부 파라미터 레지스터, 스캔 변환 제어 회로, 스캔 변환 데이터패스, 출력레지스터로 구성되며, 내부 블록도는 그림 8과 같다. 설계된 프로세서의 데이터 정밀도는 모바일 응용 분야를 고려하여 표 1과 같은 정밀도를 채택하였다. 단, X, Y 좌표의 출력 레지스터(X_out, Y_out)는 정수 필드만 구성되지만 내부에서는 정점, 에지와 스패를 올바르게 표현하기 위해 고정 소수점 형식을 갖고 있다. 내부 데이터

의 갱신 동작 과정에 표현 범위의 최대, 최소 근처에서 발생하는 덧셈, 뺄셈의 경우 오버플로우나, 언더플로우를 발생시켜 심각한 오동작을 야기 할 수 있다. 따라서 설계한 스캔 변환 프로세서는 이러한 문제를 방지하기 위해 디지털 신호 처리 프로세서와 같이 오버플로우나 언더플로우 감지시 해당 레지스터를 최대 혹은 최소 값으로 만드는 포화 연산(saturated arithmetic) 처리를 하는 기능을 지원한다.

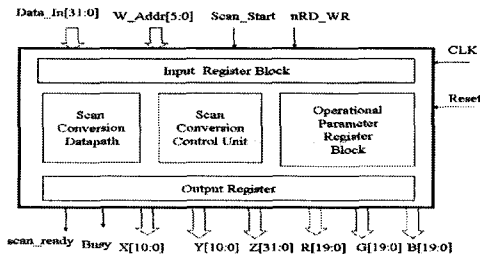


그림 8. 스캔 변환 프로세서의 블록도
Fig. 8. Block diagram for scan conversion processor

표 1. 내부 데이터의 정밀도
Table 1. Precision of internal data

내부 변수, 기울기	데이터 형식 (고정 소수점 M.N형식)
X (vertex 0, 1, 2)	12.12
Y (vertex 0, 1, 2)	12.12
Z(vertex 0, 1, 2)	20.12
R, G, B(vertex 0, 1, 2)	9.12
$\Delta x / \Delta y$ (edge 1, 2, 3)	12.12
$\Delta z / \Delta y$ (edge 1, 2, 3)	20.12
$\Delta R / \Delta y, \Delta G / \Delta y, \Delta B / \Delta y$ (edge 1, 2, 3)	9.12
$\partial z / \partial x$	20.12
$\partial R / \partial x, \partial G / \partial x, \partial B / \partial x$	9.12
X_out	11.0
Y_out	11.0

입력 레지스터는 삼각형 셋업 과정에서 계산한 기울기와 함께 정점의 속성 정보를 받는다. 파라미터 레지스터는 삼각형의 형식과 삼각형 유형 1, 2에서 필요한 2개의 삼각형 분할 처리를 지원하기 위한 정보, 스캔 정보,

삼각형의 Y 정보 등을 저장한다. 스캔 변환 데이터 패스는 제어 회로의 제어 신호에 따라 스캔 변환 동작을 수행하는 부분이다. 그림 9는 스캔의 시작 정보(x_{min})를 담고 있는 X_left 레지스터와 X의 출력 정보를 담는 Count_X 레지스터를 나타낸다. X_left 레지스터는 변처리 과정동안 에지 방향의 기울기 성분($\Delta x / \Delta y$ (edge))을 순차적으로 더해서 스캔의 왼쪽 끝 좌표를 생성하는 역할을 수행한다. Saturation 부는 오버플로우와 언더플로우 발생시 포화연산을 하는 모듈을 의미한다. X_next는 Type_1, Type_2 삼각형의 처리 시 두 번째 삼각형의 시작 스캔의 x-좌표를 갖고 있다. Count_X 레지스터의 경우 그림 6(a)에 따라 X_left 레지스터에 스캔의 시작 지점(x_{min})이 계산되면, 픽셀 채움 규칙에 따라 천장 함수(ceiling)를 통해 스캔 동작을 수행할 실제 정수 시작 좌표(x_{init})를 구한 후, 스캔의 끝에 도달할 때 까지 1씩 순차적으로 증가시키는 동작을 통해 스캔 처리를 수행한다.

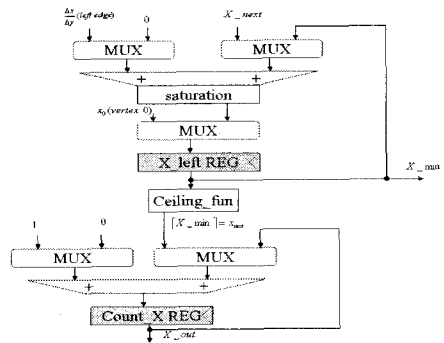


그림 9. X_left와 Count_X 레지스터
Fig. 9. X_left and Count_X register

X, Y 출력에 비해 픽셀 Z, R, G, B 성분의 경우 정수 형태의 출력이 아닌 고정 소수점 형태로 출력하도록 하여, 최근 새로운 그래픽 프로세서 기법으로 대두하고 있는 픽셀 셰이더(pixel shader)에 연결할 수 있도록 하였다. 그림 10은 붉은 색 성분, R(Red)에 대한 출력을 생성하는 보간 회로를 나타낸다. 그림 9와 다른 점은 스캔 동작시 Count_R에 x-방향의 증분($\partial R / \partial x$)성분이 순차적으로 더해진다는 점과 천장 함수(ceiling) 부분이 없다는 점이다. 다른 픽셀 성분도 유사한 방식으로 구해진다. 즉 스캔 데이터 패스에는 X, Y, Z, R, G, B 성분에 대한 보간기 회로와 스캔의 우측 끝을 지시하는 X_right 성분을 발생

시키는 보간기 회로가 존재한다. 단, Y 성분에 대한 보간기 회로는 매 스펀 라인마다 1씩 y값을 증가시키는 단순한 구조로 되어 있다. 스캔 변환 회로에 대한 제어회로는 스캔 변환 동작을 ASM(algorithmic state machine) 도표로 표현한 후, 이를 FSM(finite state machine) 하드웨어로 변환하는 기법으로 구현되었다.

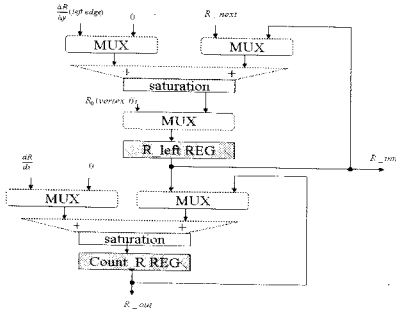


그림 10. R_left와 Count_R 레지스터
Fig. 10. R_left and Count_R register

IV. 설계 검증 및 성능 분석

본 논문에서 설계한 스캔 변환 프로세서는 먼저 개별 데이터 패스 모듈을 Verilog-HDL로 작성한 후 Modelsim 소프트웨어를 사용하여 올바른 동작을 검증하였다. 데이터 패스에 대한 동작 확인 후에 제어 회로와 통합한 전체 회로에 대한 설계 검증 환경을 구축하여 4가지 삼각형 유형에 대한 올바른 동작 수행 여부를 확인하였다. 그림 11은 Type-2 형식의 삼각형에 대한 스캔 변환 회로의 Modelsim 검증 파형을 나타낸다.

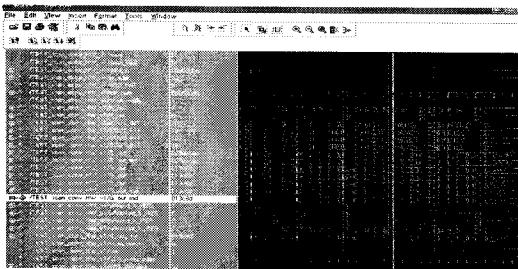


그림 11. 스캔 변환 회로에 대한 Modelsim 검증 파형
Fig. 11. Modelsim waveform for scan conversion circuit

설계된 회로는 Modelsim 검증을 마친 후, 0.35μm CMOS 표준 셀 라이브러리와 Synopsys 소프트웨어를 사용하여 합성한 결과 약 21,400개의 게이트로 구성되며 최악 전달 지연 시간은 11.37 ns로 최대 동작 주파수는 약 80 Mhz를 가짐을 알 수 있었다. 그림 12는 Synopsys 소프트웨어로 합성한 스캔 변환 회로를 나타낸다.



그림 12. 스캔 변환 회로의 합성 결과
Fig. 12. Synthesized results of scan conversion circuit

설계된 스캔 변환 회로의 경우 매 스펀마다 한번의 픽셀 채움 처리를 위한 천장 연산 동작이 필요하지만, 거의 매 클럭마다 1개의 픽셀 정보를 출력하는 구조를 갖고 있다. 따라서 예상되는 픽셀 생성율은 식 (5)와 같다.

$$pixel\ rate \approx frequency = 80\ Mpixels/sec \quad (5)$$

따라서 설계한 스캔 변환 회로는 설계 목표인 50 Mpixels/sec의 사양을 만족함을 알 수 있으며, 모바일 그래픽 응용에 효율적으로 적용 가능하다고 판단된다. 표 2는 설계한 스캔 변환 회로의 전기적 특성을 나타낸다. 본 연구에서 설계한 스캔 변환 회로는 기존 스캔 라인 스캔 변환 회로와 달리 좌변-상변 픽셀 채움 규칙을 지원함에 따라 여러 개의 삼각형이 연결된 메쉬 구조에서 공유 변을 중복 처리를 처리하지 않게 되어 성능이 향상될 수 있다.

표 2. 전기적 특성
Table 2. Electric characteristics

공정	0.35 μm standard cell
알고리즘	스캔 라인 알고리즘
픽셀 채움 규칙	Top-left filling
게이트 수	21,400
동작 주파수	80 Mhz@3.3V
최대 픽셀 생성율	80 Mpixels/sec

V. 결론

본 논문에서는 모바일 그래픽 분야에 적용할 수 있는 스캔-변환 프로세서를 설계하였다. 설계된 스캔 변환 회로는 고정 소수점 형식의 데이터 표현을 가지며, 외부에서 세 개의 삼각형 정점과 기울기 정보를 받은 후 삼각형을 네 가지 유형으로 구분하여, 변 처리와 스캔 처리를 통해 매 클록마다 한 개의 픽셀 성분(X, Y, Z, R, G, B)을 출력할 수 있다. 그리고 여러 개의 삼각형이 변을 공유하는 상황에서 공유 변상의 픽셀의 중복 처리 혹은 미처리를 방지하기 위해 좌변-상변 채움 규칙을 지원한다. 설계된 회로는 Modelsim 소프트웨어로 검증하였으며 Synopsys 소프트웨어로 합성한 결과 약 21,400개의 게이트 수로 구성되며, 최대 80 Mhz의 동작 주파수로 동작 가능하다. 스캔 변환 회로는 매 클록마다 픽셀 정보를 출력 하는 구조로, 최대 80 Mpixels/sec의 픽셀 생성율을 가지고 있기 때문에 모바일 그래픽 분야에 적용이 가능할 것으로 판단된다. 그리고 최근 주목 받고 있는 셰이더(shader) 구조의 그래픽 프로세서의 래스터라이징에 IP(intellectual property) 형태로 적용이 가능하다.

감사의 글

반도체 설계 교육센터(IDEC)의 CAD 소프트웨어 지원에 감사드립니다.

참고문헌

- [1] Masatoshi Kameyama, Yoshiyuki Kato, and Hitoshi Fujimoto, "3D Graphics LSI Core for Mobile Phone "Z3D" ," Graphics Hardware 2003, pp.60-67, 2003.
- [2] J.D. Foley, A. V. Dam, S. K. Feiner and J. F. Hughes, Computer Graphics : Principles and Practice, 2nd edition, Addison_wesley, Chapter 18, 1997.
- [3] Anders Kugler, Pixel Shading Pipelines and Display Hardware, Tubingen University, Ph.D thesis, 1999.
- [4] H-J Ackermann and C. Hornung, "The Triangle Shading Engine", Advances in Computer Graphics Hardware V, Springer-Verlag, pp.3-14, 1992.
- [5] S.R. Evans, R.L.Grimdale, P.F. Lister and A.D. Nimmo, "The AIDA Display Processor System Architecture", Advances in Computer Graphics Hardware V, Springer-Verlag, pp.15-28, 1992.
- [6] Huang, Han-Uei, Design of a VLSI scan conversion processor for high performance 3-D graphics systems, Iowa State University, Ph.D thesis, 1988.
- [7] D. Fussel and B.D. Rathi, "A VLSI-Oriented Architecture for Real-Time Raster Display for Shaded Polygons", Proceedings of Graphics Interface '82, pp.373-380, 1982.
- [8] Laszlo Szirmay-Kalos and Gabor Marton, "On the Hardware Implementation of Scan-Conversion Algorithms", 8th Symp. on Microcomputer Application, Budapest, pp.10-17, 1994.
<http://www.iit.bme.hu/~szirmay/puba.html>.
- [9] 전원호, 파이프라인 수행을 위한 타일 트래버설 기반 스캔 컨버전 유닛 설계, 연세대학교 컴퓨터과학·산업시스템공학과 석사 논문, 2001.
- [10] Juan Pineda, "A Parallel Algorithm for Polygon Rasterization," SIGGRAPH 88 Conference Proceeding, ACM Press, New York, pp.17-20, 1988.
- [11] Marcus D. Waller, Jon P. Ewins, Martin White and Paul Lister, "Efficient primitive traversal using adaptive linear edge function algorithms," Computer and Graphics, vol.23, pp.365-375, 1999.

- [12] Microsoft Inc., "Rasterization Rule", http://msdn.microsoft.com/archive/en-us/directx9_c/directx/graphics/programmingguide/GettingStarted/3DCoordinateSystems/RasterizationRules.asp, 2007, 5.
- [13] Chris Hecker, "Perspective Texture Mapping Part I : Foundations," Game Developer, pp.16-25, April/May 1995.
- [14] Abdreas Kugler, "The Setup for Triangle Rasterization," Proceeding of 11th Eurographics Workshop on Graphics Hardware '96, pp.49-58, 1996.

저자소개

최 병 윤(Byeong-Yoon Choi)



1985년 2월 연세대학교
전자공학과 졸업
1992년 8월 연세대학교
전자공학과 공학 박사

2006년 1월 ~2006년12월 오클랜드대학 방문 연구교수
1993년 3월~현재 동의대학교 교수
※ 관심분야: RISC 마이크로프로세서 설계, 그래픽 및
암호 알고리즘의 SoC 설계

하 창 수(Chang-Soo Ha)



2003년 2월 동의대학교 컴퓨터
공학과
2006년 2월 동의대학교 컴퓨터·
소프트웨어공학 석사

2006년 3월 ~현재: 동의대학교 컴퓨터응용공학 박사
과정
※ 관심분야: 그래픽 프로세서 설계, 임베디드 시스템
및 SoC 설계

Zoran Salcic



현재 뉴질랜드 University of Auckland
교수

※ 관심분야: 임베디드 시스템 및 SoC 설계