모바일 기기에 적합한 내장형 3차원 그래픽 렌더링 처리기의 저전력화

장 태 홍, 정 종 철, 우 현 재, 이 문 기
연세대학교 전기전자공학과
전화 : 02-2123-4731 / 핸드폰 : 011-565-8427

A Low Power Design of The Embedded 3D Graphics Rendering Processor for Portable Device

Tae-Hong Jang, Jong-Chul Jeong, Hyun-Jae Woo, Moon-Key Lee
Dept. of Electrical Electronics Eng., Yonsei University
E-mail : thjang@spark.yonsei.ac.kr

Abstract

This paper presents a low power design of the embedded 3D graphics rendering processor with the double span processing stage. The increase of hardware complexity by using the double span processing stage is ignorable. And the performance is equal to the rendering processor with the single span processing stage. It reduces the power consumption by using different clock frequencies.

I. 서론

휴대폰으로 대표되는 모바일 기기에 불어온 멀티미디어의 바람은 소형 카메라가 장착되어 사진을 찍을 수 있는 카메라폰, 동영상을 찍을 수 있는 캠코더폰, 고음질의 음악을 제공하는 MP3폰, 게임의 입체감을 제공하는 3D게임폰 등으로 불어오고 있다. 또한 이들 멀티미디어 기능을 통합한 SoC칩의 개발이 본격화되어 있는 상태로 휴대폰의 기본적인 선택기준은 이미 통화의 개념에서 모바일 엔터테인먼트의 개념으로 옮겨가고 있다. 고음질의 음악 제공을 위한 MP3, 게임용 3D엔진, MPEG4 디코더, 컬러 LCD 등의 다양한 멀티미디어 기능을 가진 휴대폰의 개발 추세는 휴대폰의 내부클럭을 증가시켰고 고속처리를 위한 전용의 모듈 역시 필요하게 되었다. 이러한 전용의 모듈 역시 높은 클럭속도와 더불어 높은 메모리 bandwidth를 요구하고 있다. 따라서 제한된 배터리의 용량으로 동작해야 하는 휴대폰의 소비전력 문제가 심각하게 대두되고 있다.

이에 본 논문에서는 내장형 3차원 그래픽 가속기 각 내부모듈별 전력소비를 비교하고 이에 맞게 저전력화를 위한 방안을 제시하여 모바일 기기에 적합한 내장형 3차원 그래픽 가속기를 제안한다.

II. 내장형 3차원 그래픽 렌더링 처리기

2.1 3차원 그래픽 가속기

3차원 그래픽 가속기는 실시간으로 사용자의 요구를 반영하여 영상 데이터를 축소, 확대, 회전하는 등 변형, 가공하는 장치이다. 3차원 그래픽 가속기는 영상 데이터의 크기 및 위치 변화와 빛에 대한 변화를 처리해주는 기하학 연산 처리기(Geometry Processor)와 영상 데이터에 색상 및 텍스쳐를 입히는 렌더링 처리기(Rendering Processor)로 나누어진다.

일반적인 PC와 같은 고성능의 컴퓨팅 환경에서는 기하학 처리 부분과 렌더링 처리 부분을 모두 하드웨어로 구현하고 있다. 그러나 모바일 환경에서 이 두 부분을 모두 하드웨어로 구성하는 것은 무리가 있으며

렌더링 처리기만을 하드웨어로 구성하여도 모바일 환경에서 충분한 성능을 얻을 수 있다. 따라서 이 논문에서는 위의 가정을 바탕으로 렌더링 처리 부분만을 하드웨어로 구현하였다[1][2][3].

2.2 내장형 3차원 그래픽 렌더링 처리기

구현한 내장형 3차원 그래픽 렌더링 처리기는 복잡한 연산을 피하고 모바일 환경에 적절한 성능의 그래픽 구현을 위해 고라우드 쉐이딩(Gouraud Shading) 기법을 사용하였고, 트루 컬러 및 16bit의 깊이 정보를 지원한다. 또한, 하드웨어 원근 처리 및 Bi-linear 밉매핑을 지원하며 실감 영상을 위한 투명도 비교 및 처리, 깊이 비교를 지원한다.

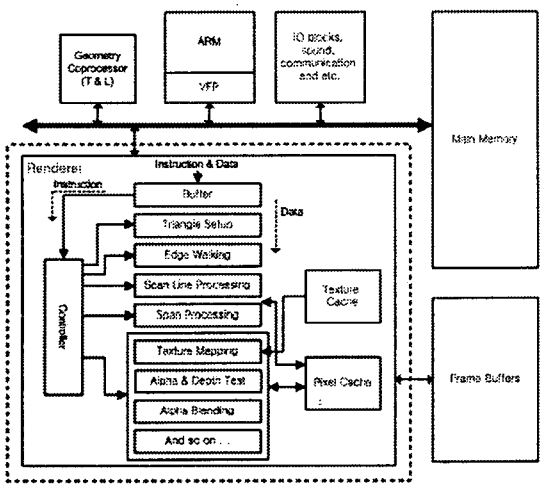

그림 1. 내장형 3차원 그래픽 렌더링 처리기

그림 1은 구현한 내장형 3차원 그래픽 렌더링 처리기가 포함된 SoC의 개념을 보여주는 것이다. ARM이 시스템 코어로 사용되며 VFP(부동 소수점 연산 처리기)이 영상 데이터 처리, 특히 기하학 연산 처리를 도와준다. ARM과 VFP에서 기하학 연산 프로그램을 이용하여 연산된 영상 데이터가 메인 메모리에 저장되고 구현한 내장형 3차원 그래픽 렌더링 처리기는 메인 메모리로부터 영상 데이터를 받아들여 처리하게 된다. 이때 처리된 데이터는 프레임 버퍼에 다시 저장되고 이후 최종 디스플레이 되게 된다.

그림에서 볼 수 있듯이 최종적으로는 구현한 내장형 3차원 그래픽 가속기를 포함해서 MPEG4 엔진, MP3, 통신 블록 등이 하나로 합쳐진 통합 멀티미디어 SoC가 탄생할 수 있을 것이다.

(1) Command parser

Command Parser는 Vertex Buffer라고 불리는 일종의 명령어와 데이터의 묶음을 받아서 명령어를 분석하고 데이터를 정렬하여 하단에서 필요한 데이터와 제어 신호를 보낸다. Vertex Buffer는 각 삼각형의 정점에서의 3차원 그래픽 데이터와 이를 처리하기 위한 제어 명령어를 가지고 있는데, 시스템 버스의 대역폭을 고려하여 크기가 고정되어 있다. 크게 제어 명령을 가지는 VB Header와 그래픽 데이터와 그 정보를 가지는 Triangle Index, Vertex Information으로 구성되어 있다.

(2) Triangle Setup

Vertex Buffer에 있는 데이터는 위치 좌표와 색상 정보를 가진 세 정점으로 이루어진 삼각형을 이룬다. Triangle Setup은 이 세 정점을 가지고 삼각형을 이루는 요소인 각 변의 기울기와 색상 정보의 증가분을 구하여 가상적인 삼각형을 만드는 과정이다. 이후의 과정이 픽셀 당 연산을 하는 반면 Triangle Setup은 폴리곤 당 한 번의 연산을 수행한다. 따라서 이 부분은 고성능의 하드웨어를 사용하기 보다는 최소한의 하드웨어를 반복적으로 사용하는 것이 더 효율적이다.

(3) Edge Walking

Edge Walking에서는 Triangle Setup에서의 가상의 삼각형으로부터 그 삼각형의 변을 따라가며 각 라인을 구하게 된다. 이때 삼각형을 둘로 나누어 연산이 수행되도록 하여 효율적인 수행이 가능하도록 하였다.

(4) Scan Line Processing

Scan Line Processing에서는 삼각형의 변을 따라가면서 삼각형 각 라인의 양 끝점을 구하여 그 데이터로부터 색상과 텍스쳐 데이터의 초기값과 그 증분을 구하게 된다.

(5) Span Processing

Span Processing은 Scan Line Processing에서 구해진 각 라인의 양 끝점의 데이터와 그 증분을 이용하여 직접적으로 삼각형 내부의 픽셀 하나하나의 데이터를 구해주는 단계이다. 색상 값은 단순히 덧셈 연산을 수행하면 되나 텍스쳐 좌표의 경우 Texture Mapping에서의 원근처리를 위하여 1/W을 구하여 텍스쳐 좌표와 곱하게 된다. 이를 위해 처리율이 1cycle인 파이프라인 나눗셈기를 사용하였다[4].

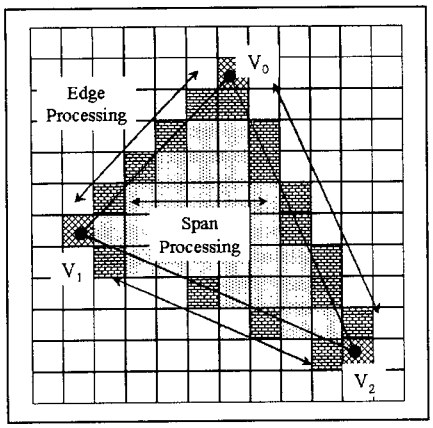

그림 2. 삼각형 내부 픽셀 데이터를 구하는 과정

(6) Texture Mapping

Texture Mapping은 이전까지의 작업으로 생성된 각 오브젝트에 텍스쳐 이미지를 입힘으로 3차원 영상 데이터에 현실감을 더해주는 과정이라고 할 수 있다.

텍스쳐 이미지를 입히는 과정에서 원근에 따라 영상이 왜곡되는 Rubber Sheet 현상이 발생하는데 이를 막기 위해서 Perspective Division을 통해 원근을 처리해주는 하드웨어를 추가하였다[5]. 또한 폴리곤의 크기에 따라 Texture Map의 크기를 바꾸어주는 Bi-linear Mip-mapping을 구현하였다.

그리고 Direct3D와 OpenGL에서 요구하는 6가지의 블렌딩 모드와 5가지의 칼라 모드를 모두 지원하며 이 과정에서 효율적으로 Texture Map 데이터를 가져오기 위해 Texture Cache를 구현하여 사용하였다.

2.3 각 모듈별 면적과 전력비교

그래픽 렌더링 처리기의 각 모듈별 면적은 표 1에서 볼 수 있듯이 각 정점의 데이터를 이용하여 각 변의 기울기와 색상값의 증분을 구하기 위해 많은 플립플롭과 나눗셈기를 포함한 Triangle Setup 단계가 가장 큰 면적을 차지했고 나머지 단계는 비슷한 면적을 차지했다.

표 2는 그래픽 렌더링 처리기의 각 모듈별 스위칭 시에 발생하는 전력의 소모를 나타내는 것으로 그래픽 하드웨어 성능에 큰 영향을 미치지 못하는 Triangle Setup이나 Edge Walking 단계가 상당히 많은 전력을 소비하는 것을 볼 수 있다. 이를 개선하기 위해 3장에서 병렬구조를 가진 Span Processing을 제안한다.

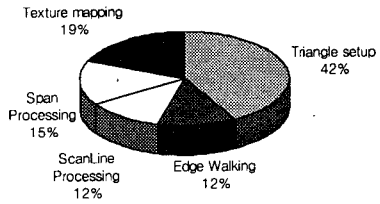

표 1. 렌더링 처리기의 각 모듈별 면적비교

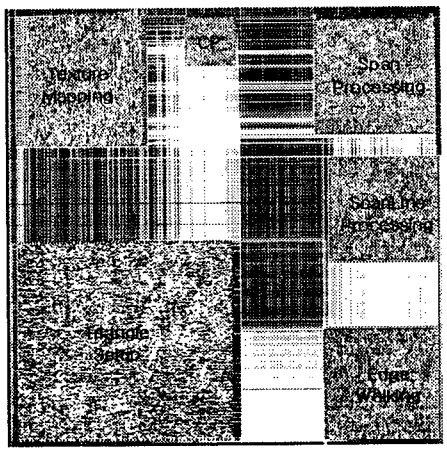

그림 3. 렌더링 처리기의 각 모듈별 합성 사진

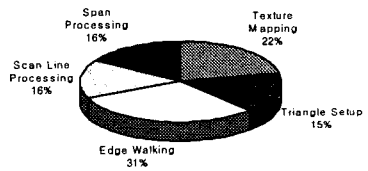

표 2. 렌더링 처리기의 각 모듈별 스위칭 전력

III. Double Span Processing Stage

3.1 필요성

Triangle Setup은 삼각형 당 1회의 수행이 이루어지고 Edge Walking은 삼각형 당 2회의 수행이 이루어진다. 반면 Scan Line Processing은 삼각형 내부의 라인 수만큼 그리고 Span Processing은 픽셀의 수만큼 수행이 이루어지게 된다. 즉 Scan Line Processing과 Span Processing이 하드웨어의 성능에 큰 영향을 미치게 되는 부분인 것이다. 따라서 Triangle Setup과 Edge

Walking에서는 반복구조를 이용하여 효율적인 하드웨어를 구현해야 하고 Scan Line Processing과 Span Processing에서는 반복구조 없이 병렬적으로 처리할 수 있어야 하는 것이다.

따라서 시작값과 증분을 이용하여 내부 픽셀 하나하나의 데이터를 구하는 Span Processing 단계에서 이를 2개의 병렬구조로 설계하여 1cycle당 2개의 픽셀을 내보내는 구조로 설계하고 성능에 큰 영향을 주지 못하는 모듈의 클럭을 낮춤으로 성능의 저하 없이 전력소모를 줄일 수 있다.

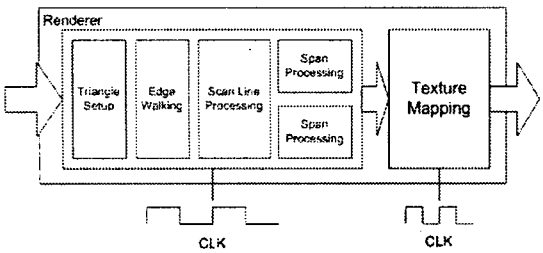

그림 4. Double Span Processing Stage와 클럭킹

3.2 성능, 면적, 전력소비의 비교

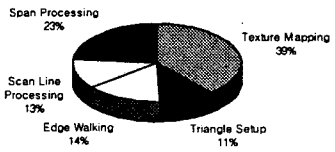

표 3. Double Span Procsseing Stage를 가진 렌더링 처리기의 모듈별 전력소모

그림 4에서 보는 것과 같이 텍스쳐 이미지가 입혀지기 전 단계에서의 최종적인 색상 데이터가 구해지는 Span Processing을 2개의 병렬구조로 하고 Texture Mapping을 제외한 앞단의 클럭을 기존 클럭의 반으로 줄이고 그에 맞게 파이프라인을 최적화 시켜 플립플롭의 수를 줄여 면적을 줄일 수 있었다. 또한 병렬화를 통한 면적의 증가는 표 1에서 보듯이 기존 렌더링 처리기에서 Span Processing이 차지하는 면적이 전체의 15%이므로 병렬화로 인한 면적의 증가는 클럭을 낮추고 파이프라인을 최적화함으로 생기는 면적의 감소와 상쇄되어 미미한 수준이 된다.

반면 Texture Mapping 단계에서는 기존의 클럭을 그대로 이용함으로 성능에서의 저하는 전혀 없도록 설계하였다.

이러한 클럭의 조절로 Texture Mapping을 제외한 3차원 그래픽 처리과정의 앞 단계에서 스위칭으로 발생하는 전력의 소비를 줄여 저전력화를 구현하였다.

Ⅳ. 결론

휴대폰으로 대표되는 모바일 환경에서 거센 디지털 컨버전스의 바람이 불고 있어 내장형 3차원 그래픽 가속기의 필요성이 더욱 강조되고 있다. 하지만 배터리의 발전이 무어의 법칙을 따르지 못함으로 해서 생기는 문제와 모바일 환경의 특징상 성능, 면적, 저전력이 모바일 환경에서의 가장 중요한 요소가 되고 있다.

이에 본 논문은 내장형 3차원 그래픽 가속기를 구현하고 이의 각 모듈별 면적과 스위칭 전력을 측정하여 성능에 영향을 주지 않으면서 부분적인 병렬구조와 이에 맞게 클럭의 조절을 통해 내장형 3차원 그래픽 가속기의 저전력을 구현했다. 앞으로 모듈별 전력소모와 내부 그래픽 처리과정 사이에서의 더욱 더 체계적인 분석으로 적절한 성능을 유지하면서 면적과 전력소모를 줄인 내장형 3차원 그래픽 렌더링 처리기를 구현할 계획이다.

참고문헌(또는 Reference)

[1] Tae-Hong Jang, Jong-Chul Jeong, Hyun-Jae Woo and Moon-Key Lee : "Embedded 3D Graphics Renderer for Mobile Devices" Proc. of KGS Winter Conference pp.471-476, Jan. 2004.

[2] Jong-Chul Jeong, Moon-Key Lee, "A Design of the Rendering Processing Unit using a Modified Midpoint Method," Proc. of KGS Winter Conference pp.337-342, Jan. 2002.

[3] 우현재, "휴대형기기에 적합한 내장형 3차원 그래픽 렌더링 처리기 설계", 연세대학교 석사 학위논문, 2003년 12월

[4] Jong-Chul Jeong, Woong Jeong, Woo-Chan Park, Tack-Don Han and Woon-Key Lee, "A New Pipeline Divider with a Small Lookup Table," Proc. of 2002 IEEE AP-ASIC, pp.33-36, Aug. 2002.

[5] Paul S. Heckbert, Henry P. Moreton, "Interpolation for Polygon Texture Mapping and Shading," State of the Art in Computer Graphics: Visualization and Modeling, pp.101-111, 1991.