OpenVG 기반 벡터 그래픽 가속기

최영[1], 홍은경[1], 이권형[1], 심용로[1], 김택규[2], 김현규[2], 오형철[3]
[1]고려대학교 대학원 전자정보공학과
[2](주)에이디칩스 연구소, [3]고려대학교(조치원) 전자및정보공학부
e-mail : ohyeong@korea.ac.kr

An OpenVG Vector Graphics Accelerator

Y. Choi[1], E.-K. Hong[1], G.-H. Lee[1], Y.-L. Shen[1], T.-G. Kim[2], H.-G. Kim[2], and H.-C. Oh[3]
[1]Dept of Electro. & Info Eng., Graduate School, Korea University
[2]R&D center, Advanced Digital Chips, [3]Dept of Elec. & Info Eng., Korea University(ChoChiWon)

Abstract

This paper presents a hardware accelerator for accelerating vector graphics applications based on the OpenVG standard. Since our design mainly targets embedded applications, we focus on efficient uses of limited resources, especially the memory bandwidth. The designed accelerator can process the images of 640x240 pixels with moderate complexity at the rate of 30 frames per second.

Ⅰ. 서론

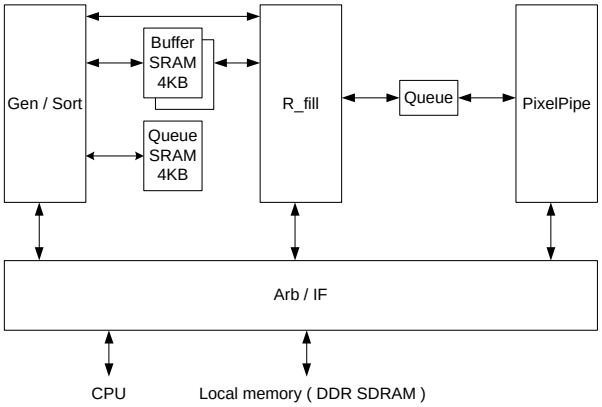
OpenVG 표준은 Flash와 같은 벡터 그래픽 라이브러리에 하드웨어 가속을 위한 인터페이스를 제공하기 위하여 정의된 표준이다[1]. 본 논문에서는 OpenVG 표준을 기반으로 하는 벡터 그래픽 응용을, 임베디드 시스템에서 효율적으로 처리하기 위하여 설계한 하드웨어 가속기를 소개한다. 설계된 가속기는 ADChips 사의 AE32KC 프로세서를 호스트 프로세서로 사용하는 SoC(system on chip)에서 사용할 것을 목표로 한다. 이와 같은 배경에서, 메모리 대역폭 등의 자원의 제약 문제의 해결 방안이 본 가속기의 설계과정 중 지속적으로 고려되었다.

Ⅱ. 가속기의 설계

본 논문에서는 OpenVG기반 그래픽의 Rendering 과정을, Application단계와 Tessellation단계, 그리고 Rasterization 및 Scissoring단계와 Pixelpipe단계의 4단계로 나누어 고려한다. OpenVG의 기준 구현(Reference Implementation(RI))과 본 논문에서 목표로 하는 응용들을 이용하여 AE32KC 프로세서에서 수행 성능을 분석한 바, 수행 시간의 대부분이 Tessellation단계의 종료이후에 소모되는 것을 알 수 있었다. 따라서 본 논문에서는 [그림 1]과 같이 Tessellation 종료 이후의 과정을 가속기에 포함시켰다. Tessellation단계까지는 호스트 프로세서에 최적화된 소프트웨어로 수행한다.

[그림 1]에 보인 바와 같이, 가속기는 크게 Gen/Sort, R_Fill, 그리고 PixelPipe단으로 이루어져 있다. 이 세 단은 크게 파이프라인 방식으로 동작하며, 각 단은 다시 파이프라인 방식으로 설계되었다. 각 단에서 사용하는 부동소수점 연산기들은 축소된 24bit 연산을 사용한다. Rasterization및Scissoring 단계는 Gen/Sort와 R_fill 두 단으로 구성되어있다. Gen/Sort단은 Active Edge 표를 만들고, R-fill단에서 효율적으로 처리될 수 있도록 Active Edge들을 정렬하는 작업을 수행한다. R_fill단에서는 Gen/Sort단에서 생성되고 정렬된 Active Edge를 이용하여 색을 나타내야할 픽셀의 좌표와 색의 농도를 결정하는 값(Coverage)을 생성 PixelPipe단에 공급한다. 이러한 데이터는 R_fill과 PixelPipe 사이의 큐를 통해 Pixelpipe에 공급된다.

PixelPipe단에서는 좌표 값과 coverage값을 이용하여 해당 픽셀을 단색으로나 그래디언트(Gradient)맵으로 또는 이미지로 채우는 역할을 하며, Blending



[그림 1] 가속기의 구조

이나 Antialiasing, Masking과 같은 효과 또한 PixelPipe단에서 이루어진다. PixelPipe단에서 생성된 픽셀의 색상 값은 Arb/IF를 통해 지역메모리로 보내진다.

메모리 대역폭 문제를 해결하기 위하여 지역메모리로의 데이터 접근에는 16bit 고정소수점 표현을 사용하였다. Gen/Sort단과 R_fill단 사이에 위치한 3개의 4KB SRAM (2개의 버퍼와 1개의 쓰기 큐)도 역시 메모리 대역폭 문제를 완화시키기 위한 것으로써, 이들을 사용하여 하나의 스캔라인이 R_fill단에서 처리되고 있는 동안에 다음 스캔라인을 Gen/Sort단에서 처리할 수 있다.

Arb/IF단은 가속기가 교착상태에 빠지지 않도록 메모리접근에 우선순위를 주고, 적시에 데이터를 공급할 수 있도록 필요한 데이터를 미리 접근하는 기능을 한다.

설계된 가속기는 크기가 640X480인 동영상을 초당 30프레임의 속도로 처리하고, 더 큰 동영상도 낮은 속도로 처리할 수 있도록 설계하였다.

Ⅲ. 구현 및 평가

설계된 가속기는 Verilog HDL를 이용하여 모델화하고, Cadence NC-verilog를 이용하여 기능을 검증하였으며, 0.18um CMOS 표준 셀 라이브러리를 이용하여 합성하였다. 모의실험과 합성 과정에서 Synopsys DesignWare 라이브러리를 사용하였다.

설계된 가속기는 3개의 SRAM 모듈과 Arb/IF단을 제외하고 대략 25만 등가게이트의 비용으로 구현할 수 있다. 각 4KB SRAM의 구현에는 약 17500 등가게이트 정도의 비용이 든다. Arb/IF unit의 비용은 가속기가 사용될 시스템에 의존한다.

가속기의 수행시간은 이미지의 복잡도에 큰 영향을 받는다. [그림 2]는 검증에서 고려된 동영상의 예를 보인 것으로서, 하나의 꽃만 움직이는 영상이다.

설계된 가속기는 640X480 크기의 비교적 복잡하지 않은 동영상(Blending 기법을 사용하고, 스캔라인 당 Active Edge의 수가 100이하인 동영상)을 초당 30프레임의 속도로 처리할 수 있다. 또한 같은 정도로 복잡한 1024X768 크기의 동영상을 초당 18 프레임의 속도로 처리할 수 있다.

Ⅳ. 결론

본 논문에서는 OpenVG기반 그래픽의 Rendering 과정을 4개 단계로 나누었을 때, Rasterization과 PixelPipe단계의 처리를 가속하기 위한 하드웨어 가속기를 설계하였다. 설계된 가속기는 비교적 복잡하지 않은 640x480 크기의 동영상을 초당 30 프레임의 속도로 처리할 수 있으며, 메모리 대역폭 등 자원이 제약된 내장형시스템에서 벡터 그래픽 응용의 처리에 도움이 될 것으로 기대된다.

본 논문에서는 Tessellation단계를 배제하였으나 이 단계의 처리 결과를 뒷 단계에서 사용하는 만큼, 가속기 내에 함께 효율적으로 구현함으로써 특히 메모리 대역폭 등의 문제를 완화시킬 수 있다. 현재, 하드웨어 비용이 허용되는 응용을 대상으로, Tessellation단도 포함하는 가속기의 개발을 진행하고 있다.

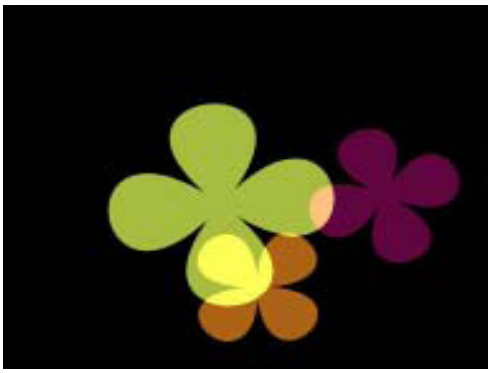
감사의 글

본 연구는 (주)에이디칩스와 반도체설계교육센터(IDEC)의 지원을 받아 수행되었습니다.

참고문헌

[1] Khronous Group std., *OpenVG*, Khronous Group Standard for Vector Graphics Accelerations, www.khronous.org., 2005.

[2] http://www.khronos.org/openvg/



[그림 2] 검증에 사용된 동작 이미지의 예