

그래픽스 전용 메모리 설계

김성진[†] · 문상호^{**}

요 약

본 논문에서는 컴퓨터 그래픽스 연산의 메모리 액세스 속도를 개선하는 새로운 메모리 구조를 갖는 그래픽스 전용 메모리(SMGRA, Special Memory for GRAPhics)를 제안한다. 제안된 그래픽스 전용 메모리는 사각형 영역의 화소를 동시에 처리할 수 있는 Whelan이 제안한 장방향 어레이 메모리 구조를 개선하여 주소 디코딩 시간을 줄여주고 멀티플렉싱 기법을 사용하여 주소핀 수를 1/2로 줄인다. 또한, 그래픽스 전용 메모리는 간단한 연산 로직을 가지므로 3차원 그래픽스 처리시 요구되는 읽기-수정-쓰기 메모리 사이클을 쓰기 메모리 사이클로 대체하므로 프레임 버퍼 액세스 속도를 개선한다.

Special Memory Design for Graphics

Sung-Jin Kim[†], Sang-Ho Moon^{**}

ABSTRACT

In this paper, we propose a Special Memory for Graphics(SMGRA) which accelerates memory access time for graphics operations. The SMGRA has a rectangular array memory architecture which has already proposed by Whelan to process pixels in the rectangle area simultaneously, but the SMGRA should improve address decoding time and reduce the number of address pins by using address multiplexing scheme. The SMGRA has a Z-value comparator in the DRAM which is to convert read-modify-write Z buffer into single-write only operation that improves approximately 50% frame buffer access bandwidth.

1. 서 론

최근 여러 분야에서 멀티미디어의 활용이 급속히 확대되어 컴퓨터 그래픽스의 응용이 더욱 활성화되어 가고 있다. 이와 같은 컴퓨터 그래픽스의 처리 결과는 많은 비트맵 데이터로 구성되므로 프레임 버퍼의 메모리 액세스 대역폭 향상이 전체 그래픽스 처리의 성능 개선에 매우 중요한 역할을 한다.

컴퓨터 그래픽스에서는 프레임 버퍼 대역폭을 개선하기 위하여 메모리의 구성 방법 및 구조에 대한 여러 가지 방법들이 제안되었는데 크게 두 가지로 요약될 수 있다[1,2]. 첫 번째는 메모리 구조를 수정하여 동시에 여러 개의 메모리 셀과 열이나 사각형 영역으로 구성된 블럭 단위로 연산을 가능하게 하여 전체적인 메모리 액세스 시간을 줄이는 방법이고, 두

번째는 프로세싱 소자를 메모리에 결합시켜 프로세서와 메모리사이의 데이터 전송량을 줄이는 방법이다. 전자의 경우는 지금까지 두 개의 포트를 가진 VRAM(Video RAM), 클락에 동기를 맞추어 고속의 메모리 액세스가 가능한 SDRAM (Synchronous DRAM), RDRAM(Rambus DRAM), 윈도우 연산을 효율적으로 처리하는 WRAM(Window RAM) 및 Whelan의 장방향 어레이 메모리 등이 있다. 후자의 경우는 SLAM(Scan Line Access Memory)[3], EMC(Enhanced Memory Chip)[4]와 FBRAM[5] 등이 있다.

한편, 기존에 제안된 여러 가지 방안들은 블럭 단위 구조를 가지므로 2차원 그래픽스 연산에는 적합하지만 3차원 그래픽스 연산에는 부적절하고, EMC와 같이 하나의 화소당 하나의 프로세싱 소자를 적용함으로써 구현 비용이 비싼 문제 등이 있었다. 3차원 컴퓨터 그래픽스의 경우, 블럭 단위의 연산도 필요하지만 화소의 색상값을 결정하기 위한 계산 비용도

[†] 연암공업대학 컴퓨터정보기술과

^{**} 영남이공대학 컴퓨터정보기술과

상당하므로 위의 두 가지 방법을 절충하는 것이 바람직할 것이다.

본 논문에서는 기본적인 DRAM의 특성인 대용량의 메모리를 가지면서 장방형 어레이(rectangular array) 메모리 구조를 가져 대부분의 컴퓨터 그래픽스 응용 프로그램에서 많이 사용되는 화면 지우기와 동일한 색상값으로 사각형 채우기와 같은 블럭 단위 연산을 효과적으로 처리하고, 내장된 간단한 비교 연산 로직을 이용하여 많은 메모리 액세스 연산을 필요로 하는 3차원 그래픽스 연산을 가속할 수 있는 그래픽스 전용 메모리(Special Memory for GRaphics, SMGRA)를 제안한다.

제안된 그래픽스 전용 메모리는 기존의 Whelan이 제안한 장방형 어레이 메모리[6,7]의 디코딩 시간을 줄여주고 주소 멀티플렉싱 기법을 사용하여 주소핀 수를 1/2로 줄여 하드웨어 구현을 간단하게 한다. 또한 간단한 비교 연산 장치를 추가로 가지므로 그래픽스 처리시 많은 메모리 액세스 시간과 계산 비용이 소모되는 3차원 그래픽스 알고리즘을 효과적으로 지원한다. 그래픽스 전용 메모리 구조는 임의의 모양을 갖는 사각형 영역의 하위 왼쪽 구석(lower-left corner)과 상위 오른쪽 구석(upper-right corner)의 열 주소와 행 주소를 받아서 이 영역에 속하는 여러 개의 열과 행을 동시에 선택하고 선택된 영역내의 모든 화소들에 대해 색상값을 동시에 지정할 수 있으며, 간단한 비교 연산자를 이용하여 전역 Z 버퍼 알고리즘을 메모리 내부에서 수행하므로 프레임 버퍼 액세스를 가속화 한다.

2. 그래픽스 전용 메모리의 구조 및 응용

그림 1에서 보여주는 SMGRA는 장방형 어레이 메모리, Z값 비교 연산 로직, 그리고 제어 부분으로 구성된다. SMGRA는 임의의 크기를 갖는 사각형 영역의 화소들에 대하여 한 번의 쓰기 연산만으로 동시에 값을 지정할 수 있어 그래픽스 연산에서 자주 사용되는 화면 지우기 기능을 쉽게 처리할 수 있으며, 또한 선 그리기와 장방형 영역 채우기(rectangular filling) 연산들도 기존의 메모리 구조보다 빠른 시간 내에 처리할 수 있다. 그리고 비교기로 구성된 연산 장치를 이용하여 전역 Z 버퍼 알고리즘을 처리하기 위해 Z값 비교 연산을 메모리 내부에서 처리함으로

써 읽기-수정-쓰기 메모리 사이클을 단순한 쓰기 사이클로 대체하므로 프레임 버퍼 액세스 속도를 향상 시키게 된다.

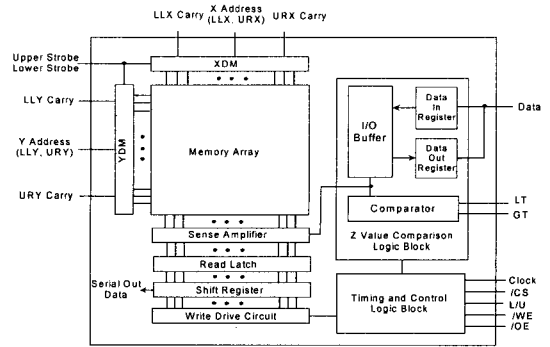


그림 1. SMGRA의 상세 블럭도

2.1 장방형 어레이 메모리의 특성

장방형 어레이 메모리 구조에서 선 그리기와 다각형 채우기(polygon filling)에 관한 알고리즘 및 이점들은 Whelan이 제안한 방법에 잘 나타나 있으며, 여기서는 0~45° 사이의 선 그리기에 관해서만 살펴본다.

일반적인 1-화소 액세스 메모리에서의 메모리 액세스 시간을 t_p , 장방형 어레이 메모리에서의 액세스 시간을 t_a 라 하고, 그리고자 하는 선의 양 끝점들 간의 x 좌표값과 y 좌표값의 차이를 그림 2와 같이 각각 dx와 dy라 하자. 그림 2 a)에 보인 선을 선 그리기의 대표적인 Bresenham 알고리즘[8]을 사용하여 그리는 경우 1-화소 액세스 메모리를 사용하면 $\max(|dx|, |dy|) + 1$ 번의 쓰기 연산을 필요로 하고, 액세스 시간은 $t_p \times (\max(|dx|, |dy|) + 1)$ 이 된다. 반면, 그림 2 b)에 보인 것과 같이 장방형 어레이 메모리를 사용하여 사각형들로 선을 그리는 경우에는 min

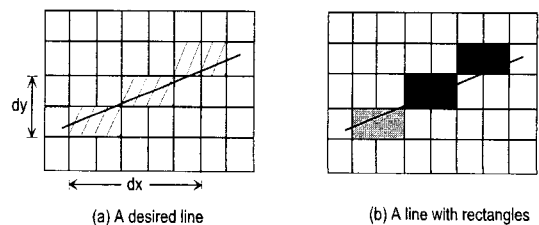


그림 2. 사각형으로 선 그리기

($|dx|, |dy|$)+1번의 쓰기 연산이 필요하며, 메모리 액세스 시간은 $t_a \times (\min(|dx|, |dy|) + 1)$ 이 된다. 따라서, 장방형 어레이 메모리의 액세스 시간 t_a 가 기존의 1-화소 액세스 메모리의 액세스 시간 t_b 보다 크지 않다면 장방형 어레이 메모리는 기존의 메모리에 비해 높은 성능을 가질 수 있다.

메모리에 대한 액세스 시간은 메모리 내부에서의 주소 디코딩 시간과 데이터를 저장하는 메모리 셀 액세스 시간으로 나눌 수 있으며, 앞의 두 가지 구조에서 같은 메모리 셀 구조를 사용한다면 셀 액세스 시간은 같게 되고 주소 디코딩 시간에 의해 t_b 와 t_a 의 값이 결정된다. 장방형 어레이 메모리 구조는 기존의 메모리와 같은 메모리 셀 구조를 사용하고 주소 디코딩 로직은 기존 메모리에서의 주소 디코딩 시간과 거의 같은 디코딩을 가지므로 t_b 와 t_a 의 값은 거의 같게 된다. 따라서, 장방형 어레이 메모리는 일반 메모리에 비해 빠르게 선을 그릴 수 있으며, 다각형 채우기 연산의 경우에 $O(n)$, 박스 채우기 연산의 경우는 $O(n^2)$ 의 성능향상 효과를 갖는다.

2.2 장방형 어레이 메모리의 구조

일반적인 DRAM 구조에서 1-비트 데이터를 저장하기 위한 메모리 셀의 구조는 1개의 트랜지스터와 1개의 커패시터로 구성되며, 열 선택선이 작동되면 읽기/쓰기 신호에 따라 커패시터에 저장된 데이터값이 데이터선에 실린다. DRAM은 한 사이클에 한 개의 메모리 셀에 데이터를 읽거나 쓸 수 있는 반면에 여러 개의 열들과 행들을 동시에 선택하기 위해서는 Whelan에 의해 제안된 밴디드(banded) 주소 디코딩 개념을 도입하기로 한다. 그림 3에 보인 것과 같이 화면상의 사각형 영역은 하위 열 주소(Lower Left Y, LLY), 상위 열 주소(Upper Right Y, URY), 하위

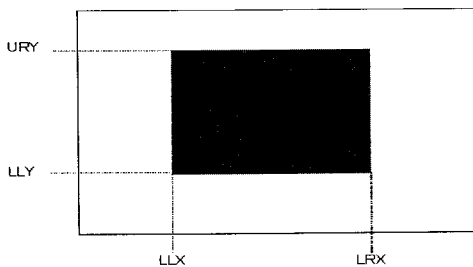


그림 3. 사각형 영역 채우기

행 주소(Lower Left X, LLX)와 상위 행 주소(Upper Right X, URX)로 표현된다. 이러한 사각형 영역이 동일한 색상값으로 표현되기 위해서 DRAM의 경우에는 $n \times m$ ($n=URX-LLX+1, m=URY-LLY+1$)번의 메모리 쓰기 사이클이 필요하다. $n \times m$ 번의 메모리 쓰기 사이클을 한 번의 메모리 쓰기 사이클로 대체함으로써 사각형 영역 채우기 동작을 빠르게 처리해 줄 수 있는 메모리 구조가 Whelan에 의해 제안되었고, 열 선택선과 행 선택선을 갖는 메모리 셀들로 이루어진 메모리 어레이, 하위 열 주소부터 상위 행 주소까지의 n 개의 행들을 동시에 선택할 수 있는 XBD(X Banded Decoder)와 하위 열 주소부터 상위 열 주소까지 m 개의 열들을 동시에 선택할 수 있는 YBD(Y Banded Decoder) 등으로 이루어진다.

XBD와 YBD는 같은 주소 디코딩 로직으로 구성되며, 그림 4에 보인 바와 같이 하위 주소의 상위 주소들과 상위 주소의 하위 주소들은 논리적인 AND로 결과를 얻는다.

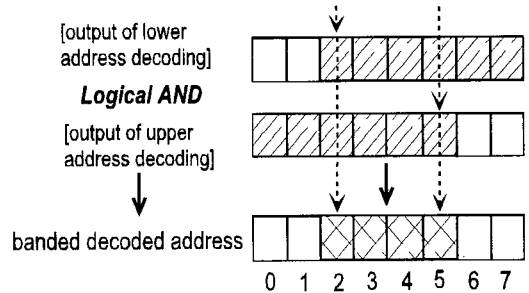


그림 4. 하위 주소가 2이고 상위 주소가 5인 경우 밴디드 주소 디코딩

2.3 SMGRA의 장방형 어레이 메모리

그림 5는 Whelan이 제안한 장방형 어레이 메모리에서 2개의 상위 주소(A_{OU})와 2개의 하위 주소(A_{OL})를 갖는 밴디드 주소 디코더 로직을 보여주고 있다. 이러한 구조는 전파지연에 의해 n 개의 주소선에 대하여 $O(\log n)$ 의 디코딩 시간을 가지며, X (행) 주소선의 수와 Y (열) 주소선의 수가 다른 경우, 주소선의 수가 많은 쪽의 디코딩 시간에 의하여 메모리 사이클 시간이 결정되게 된다.

또한, 이러한 구조는 상위 주소와 하위 주소를 함께 공급하므로 n 개의 주소선에 대하여 $2 \times n$ 개의

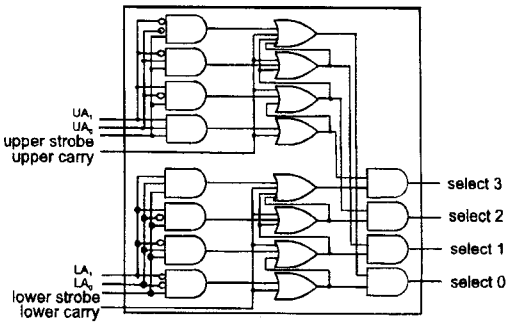


그림 5. 전파지연 체인을 가지는 Whelan의 밴드 주소 디코더

주소핀이 필요하다. i 개의 X 주소선과 j 개의 Y 주소선을 갖는 메모리의 경우에 필요한 주소핀 수는 $2 \times i + 2 \times j$ 가 되어 실제 메모리 제조시 주소핀 수가 많아지는 단점이 있다.

본 논문에서는 2차원 그래픽스 연산을 위해서 SMGRA는 Whelan이 제안한 장방향 어레이 메모리에서의 밴드 디코더의 주소 디코딩 시간을 n 개의 주소선에 대하여 $O(1)$ 으로 줄이고, 주소 멀티플렉싱 방법에 의해 n 개의 주소선에 대하여 n 개의 주소핀만 사용하도록 주소핀 수를 줄이는 방안을 제안한다. SMGRA의 장방향 어레이 메모리에서는 그림 6에서 보인 바와 같이 순차적으로 입력되는 하위 주소와 상위 주소를 선택하여 디코딩하는 XDM(X Decoder/Multiplexer)와 YDM(Y Decoder/Multiplexer), 앞단 디코더/멀티플렉서의 출력을 저장하여 원하는 밴드에 해당하는 선택선들을 선택하여 주는 X 셀렉터와 Y 셀렉터로 구성된다. L/U 선택 신호는 디코더의 출력을 저장하기 위해 사용되는 신호이다. SMGRA의 장

방향 어레이 메모리에서 사용된 XDM은 YDM과 같은 구조를 가지며, X 셀렉터도 Y 셀렉터와 같은 구조를 가진다.

그림 7에서는 SMGRA의 장방향 어레이 메모리에서 사용된 디코더/멀티플렉서와 셀렉터로 구성된 주소 디코딩 로직의 개념도를 보였다. SMGRA에 사용된 주소 디코딩 로직은 기존의 밴드 디코더에 비해 OR 게이트의 출력이 다른 OR 게이트의 입력에 연결되지 않고, 주소 디코딩 결과가 입력에 동시에 연결되므로 기존의 방법 보다 빠른 밴드 주소 디코딩 시간을 갖는다. 즉 Whelan의 방식에서는 메모리 주소 디코딩 시간이 그림 5에서와 같이 선택되는 밴드의 크기에 따라서 통과해야하는 OR 게이트의 수는 비례하여 증가하므로, OR 게이트의 전파 체이닝(chaining)의 수에 따라서 메모리 주소 디코딩 시간은 $O(\log n)$ 이 된다. 그러나 SMGRA의 경우는 선택되는 밴드의 크기와 무관하게 항상 같은 메모리 주소 디코딩 시간이 소요되므로 $O(1)$ 로 향상된다.

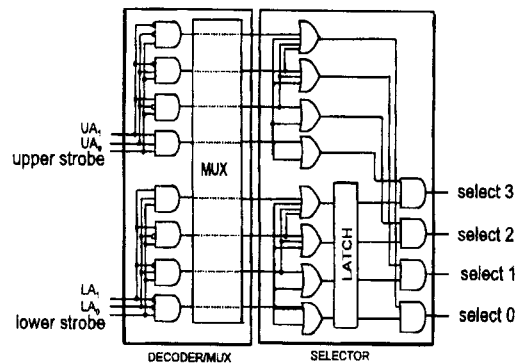


그림 7. SMGRA의 주소 디코딩 로직의 개념도

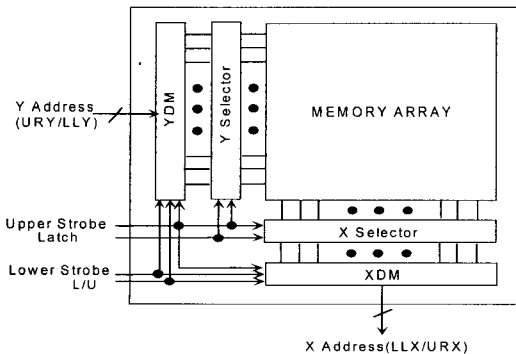


그림 6. SMGRA의 주소 디코딩 블록도

SMGRA의 주소 디코딩 로직의 동작은 다음과 같다. 먼저, 하위 주소 [LA₁, LA₀]을 공급하고, 아래 스트로브 신호를 구동하여 셀렉터 안의 OR 게이트는 하위 주소에 대한 하위 밴드 주소 디코딩 값을 디코더로부터 공급받아 내부의 래치에 저장한다. 상위 주소 [UA₁, UA₀]에 대해서도 같은 방식으로 상위 주소에 대한 상위 밴드 주소 디코딩 출력을 만들어 준다. 이 값들은 AND 게이트에 의해 이전에 래치에 저장된 하위 주소에 대한 밴드 주소 디코딩 값과 AND되어 원하는 밴드에 해당하는 선택선들을 구동하게 된다.

2.4 SMGRA의 비교 연산 장치

Whelan에 의해 제안된 장방향 어레이 메모리 구조는 선 그리기나 다각형 채우기와 같은 2차원 그래픽스 연산들을 빠르게 처리할 수 있지만, 화소마다 Z값의 계산 결과에 따라 화소의 색상값 데이터를 저장하는 3차원 그래픽스 연산들을 지원하지는 못한다.

그러나 SMGRA는 그림 1과 같이 장방향 어레이 메모리에 간단한 Z값 비교기를 추가하여 3차원 그래픽스 처리에 필요한 Z값 연산을 SMGRA 내부에서 수행되게 하여 프레임 버퍼 액세스 대역폭을 개선한다. 3차원 그래픽스 처리를 위해 기존의 Z값과 새로 계산된 Z값을 비교하여 관찰자의 위치와 더 가까이 있는 물체의 화소값을 프레임 버퍼에 저장하기 위해 읽기-수정-쓰기의 메모리 사이클이 요구된다. 그러나 SMGRA에 내장된 Z값 비교기를 이용하여 단순한 쓰기 메모리 사이클로 대체하므로 메모리 액세스 속도를 향상시킨다.

고집적도의 대용량 DRAM의 장점을 기본으로 이용하여 개발된 여러 가지 다양한 메모리는 2차원 그래픽스 처리에는 적합하지만 3차원 그래픽스 처리에는 적합하지 못함을 알 수 있다. 따라서, 대용량 DRAM의 장점을 살리면서 3차원 그래픽스 처리시에 메모리 액세스 병목 현상에 직접적으로 영향을 미치는 그래픽스 연산을 메모리 내부에서 처리되게 하여 메모리 병목 현상을 제거하는 방안을 제안한다. 즉, 3차원 그래픽스 연산에서 메모리 액세스에 매우 연관이 많은 그래픽스 알고리즘을 메모리 내부에 추가하여 별도의 소자로 처리함으로써 그래픽스 처리의 성능을 향상시킨다.

2.4.1 SMGRA를 이용한 시스템 구조

SMGRA는 3차원 그래픽스 처리에 필요한 Z값 연산을 메모리 칩 내부에서 수행하므로 프레임 버퍼 대역폭을 개선한다. SMGRA를 이용한 시스템 구조는 그림 8에 나타나 있다. 프레임 버퍼는 색상값을 저장하는 색상 버퍼(color buffer), 메뉴나 배경색을 저장하는 오버레이 버퍼(overlay buffer), 그리고 Z값을 저장하는 Z 버퍼(Z buffer)로 구성된다. Z 버퍼링을 위하여 SMGRA는 새로 계산된 Z값과 기존의 Z값과 비교하여 더 크면 GT 신호를 발생하고 더 작으면 LT 신호를 발생시킨다. 이 신호가 래스터라이저(rasterizer)로 전달되면 래스터라이저는 이것을

이용하여 Z 버퍼 내용을 그대로 유지할 것인지 혹은 갱신할 것인지를 나타내는 /WE 신호를 WE 디코딩 로직에서 발생시킨다[9,10].

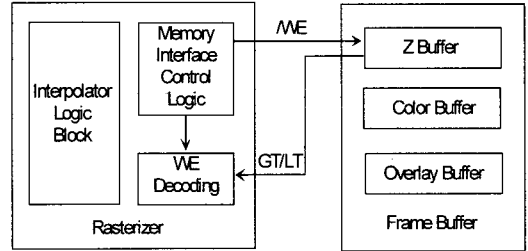


그림 8. SMGRA를 이용한 래스터라이저와 프레임 버퍼 블록도

2.4.2 SMGRA의 Z값 비교기 구조

SMGRA의 Z값 비교기 구조는 그림 1에서 보여주는 것과 같이 Whelan의 RAM(Rectangular Array Memory) 구조와 액세스 사이클에 큰 변화가 없도록 하고 RAM 내부에 단위 메모리의 입출력 비트(데이터 선) 수에 해당하는 비교기와 비교한 결과를 외부의 제어 로직으로 전달할 수 있는 GT와 LT의 두 개 신호를 추가로 가진다. 그리고 이들 신호로부터 비교한 결과 정보를 받아 메모리의 쓰기 신호를 디코딩해 주는 외부 제어 로직으로 구성된다.

2.4.3 SMGRA의 Z값 비교기의 동작

SMGRA의 비교 동작 사이클은 기존 DRAM에서 제공하는 쓰기 사이클을 변형하여 구현한다. DRAM의 늦게 쓰기 사이클(late write cycle)은 출력 인에이블로 조정되는 쓰기 사이클(OE(Output Enable) controlled write cycle)이라고도 한다[11]. 이 방법에 의한 메모리 쓰기 사이클은 열 주소를 스트로우브함과 동시에 데이터를 쓰는 조기 쓰기 사이클(early write cycle) 동작과는 달리, 주소 지정에 의한 메모리 칩 내의 셀 지정이 끝난 후에 쓰는 동작이 이루어진다. DRAM은 메모리 셀 어레이의 주소가 지정되면 그 주소에 해당하는 메모리 셀 어레이에 저장되어 있던 데이터가 출력 버퍼 전달까지 전달되는 특성 때문에 늦게 쓰기 사이클에서는 OE 신호를 이용하여 메모리 셀로부터 나온 데이터가 버퍼를 통하여 칩 외부로 출력되는 것을 막아 주도록 되어 있다. SMGRA는 이러한 DRAM의 늦게 쓰기 사이클 특성

을 이용하여 그림 9와 같이 L/U 선택 신호에서 U 선택 신호가 인에이블되면 주소가 지정된 셀의 데이터는 출력 버퍼 전달까지 전달되어 진다.

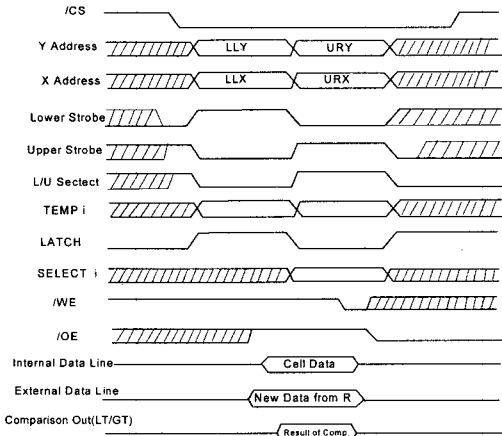


그림 9. SMGRA의 Z값 비교 연산 타이밍도

한편 새로운 데이터도 U 선택 신호가 인에이블되는 시점과 동시에 메모리로 전달된다. 입출력 데이터선은 칩 내부 비교기의 입력단에 직접 연결되어서 셀에서 나온 데이터와 비교하여 그 결과를 래스터라이저 내의 메모리 제어 로직으로 출력한다. 각 메모리의 단위 칩으로부터 래스터라이저로 입력된 비교 결과는 래스터라이저 내의 로직으로 입력되어 /WE 신호를 발생시킬 것인지의 여부를 결정하는 메모리 제어신호 발생 로직으로 전달된다. 그림 10은 8비트 데이터 선을 갖는 SMGRA로 24비트 Z 버퍼를 구성할 경우 래스터라이저에 내장되는 메모리 쓰기 신호 제어 로직을 보여준다. 기존의 Z값과 새로 계산된 Z값을 비교하여 더 크면, 즉 관찰자의 눈에 더 가까이 있는 물체의 Z값에 대해 GT 신호를 발생하여 쓰기 신호인 /WE 신호를 발생시킨다.

3. 성능분석 및 시뮬레이션

그래픽스 시스템의 성능은 여러 가지 요소에 의해 결정되지만[12], 여기서는 그 중에서도 가장 심각한 병목현상의 원인이 되는 프레임 버퍼 대역폭과 가장 밀접한 관련이 있는 메모리 액세스 속도를 기존의 메모리와 SMGRA를 비교한다. 먼저 단순히 메모리

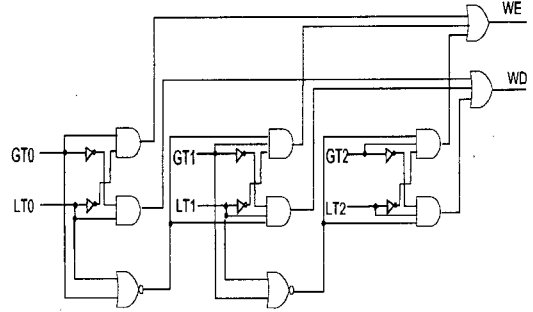


그림 10. Z값 비교 연산 결과에 따른 메모리 쓰기 인에이블(WE) 로직

액세스의 속도만을 비교하기 위해 2차원 그래픽스 연산에서 가장 중요한 선 그리기와 사각형 채우기에 대해서 기존의 메모리와 SMGRA를 비교하면 다음과 같다. 1-화소만의 액세스시는 앞서 살펴본 일반적인 DRAM과 Whelan 방식은 t_a 로 메모리 액세스의 성능이 거의 비슷하고 10개의 화소로 구성된 여러 가지 형태의 선들을 그릴 때의 성능은 선의 형태에 따라서 메모리 액세스 효율이 다르다.

한 개의 메모리칩 내에서 그려질 때와 두 개의 메모리칩에 $\frac{1}{2}$ 씩 걸쳐서 그려질 때를 비교하면, 일반적인 DARM의 경우는 페이지 모드를 지원할 수 있는 수평선의 경우 최대의 메모리 액세스 효율을 얻을 수 있는 반면에 대각선의 경우는 메모리 액세스 효율이 가장 낮다. 그러나 Whelan 방식과 SMGRA의 경우는 대각선을 제외하고는 선의 형태와 무관하게 일정한 메모리 액세스 효율을 발휘함을 알 수 있다[7]. 또한 가로(m) × 세로(n)가 100 화소로 구성된 사각형 채우기를 수행할 때의 성능을 비교하면 SMGRA의 장점이 최대한 발휘됨을 알 수 있다. 그림 11은 메모리 액세스 성능을 비교하는 다양한 형태의 선과 사각형을 보여준다.

일반적인 DRAM과 장방형 메모리의 주소 디코딩 시간은 열과 행의 주소 디코딩을 각각 순차적으로 하느냐 동시에 하느냐에 근본적인 차이가 있고 내부의 구체적인 구조에 따른 시간 차이는 미세한 것이므로 무시할 수 있다. Whelan 방식에서는 선택되는 밴드의 크기, 즉 그리고자 하는 선이나 사각형의 크기에 따라서 OR 게이트의 전파 체이닝 시간이 다르다. Whelan 방식에서 OR 게이트 전파 지연 시간이 g_d 일 때 주소 디코딩 시간은 ($g_d \times$ 최대 밴드의 크기)가 되

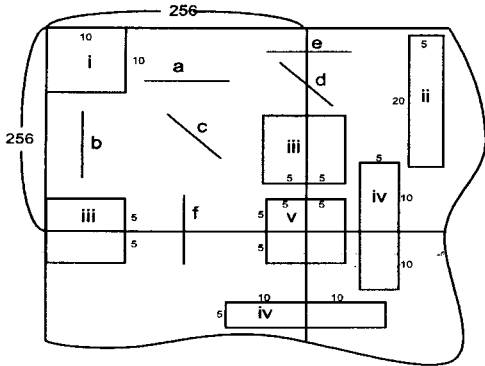


그림 11. 다양한 형태의 선과 사각형

표 1. 100개의 화소로 구성되는 다양한 사각형 채우기 성능 비교

	DRAM	Whelan 방식	SMGRA
정사각형(m=10, n=10)이 메모리 내에 위치(i)	$(t_a + 0.3 \times 9t_a) \times 10$	$t_a + 10g_d$	$t_a + g_d'$
직사각형(m=20, n=5)이 메모리 내에 위치(ii)	$(t_a + 0.3 \times 19t_a)$	$t_a + 20g_d$	$t_a + g_d'$
정사각형이 두 개의 메모리 경계에 위치(iii)	$(t_a + 0.3 \times 4t_a) \times 10$	$t_a + 10g_d$	$t_a + g_d'$
직사각형이 두 개의 메모리 경계에 위치(iv)	$(t_a + 0.3 \times 9t_a)$	$t_a + 10g_d$	$t_a + g_d'$
정사각형이 4개의 메모리 경계에 위치(v)	$(t_a + 0.3 \times 4t_a)$	$t_a + 5g_d$	$t_a + g_d'$

고, SMGRA의 경우 주소 디코딩 시간은 밴드의 크기와 무관하게 항상 일정한 시간 $t_a + g_d'$ 를 갖는다. 여기서 g_d' 는 SMGRA에서 주소 디코딩을 위해 별도로 추가된 로직의 게이트 통과지연 시간이다. 즉, 그림 7에서 보여주듯이 SMGRA에서 주소 디코딩 시간은 Whelan 방식에서 게이트 통과지연시간 $g_d = ((AND + MUX + OR) + (AND + MUX + OR) + AND)$ 와 SMGRA에서만 별도로 추가 되는 게이트가 $(OR + 2 MUX + LATCH)$ 이므로 $g_d' = g_d + (OR + 2 MUX + LATCH)$ 이다. VLSI 내부에서 각 게이트의 통과지연 시간을 2~3ns로 가정할 때 Whelan 방식에 비해 약 14~

21ns만 추가된다. DRAM의 경우 수평선을 그릴 때 메모리의 정규 읽기나 쓰기 모드 보다 훨씬 빠른 페이지 모드를 이용할 수 있으므로 10개의 화소에 대해 $t_a + 0.3 \times 9t_a$ 가 유도되어 이 경우 메모리 액세스 속도가 약 70% 정도 향상된다. 표 1에서 알 수 있듯이 Whelan의 방식과 SMGRA의 경우는 그리고자 하는 선의 길이가 길고 사각형의 크기가 클수록 더욱더 빠르게 처리할 수 있다.

한편, 액세스 시간이 50ns인 DRAM을 기준으로 메모리 액세스 시간 t_a 는 약 90ns가 되므로 장방형 메모리의 경우 기존의 DRAM과 비교할 때 가장 좋은 성능은 가장 긴 선 그리기와 사각형 채우기에서 밴드의 크기가 256인 경우이므로 같은 장방형 메모리인 SMGRA와 Whelan 방식을 비교하면 $(t_a + 256 \times g_d) / (t_a + g_d')$ 의해 SMGRA는 최대 약 4~5배의 성능 향상을 얻을 수 있다.

3차원 그래픽스 처리시는 그래픽스 전용 메모리 내부의 Z 값 비교가를 이용하므로 읽기-수정-쓰기 메모리 사이클을 단순한 쓰기 사이클로 대체하므로 프레임 버퍼 액세스 속도를 가속화하여 메모리 액세스 시간을 개선할 수 있다. 따라서 Z 버퍼 알고리즘을 이용한 3차원 그래픽스 처리시 읽기-수정-쓰기 사이클을 쓰기 사이클로 대체하므로 래스터라이저와 프레임 버퍼 사이의 액세스 대역폭을 개선할 수 있다.

SMGRA의 3차원 그래픽스 처리의 성능을 비교하기 위해 주전자와 공의 이미지를 생성할 때에 기존의 DRAM을 이용한 경우와 성능의 비교가 표 2에 나타

표 2. 3차원 이미지 생성시 프레임 버퍼 액세스 속도 비교

			DRAM	SMGRA	비교
주전자	삼각형 수	3729	$(t_{rwc} + 0.5 \times 5t_{rwc}) \times 13 \times 3729$	$(t_p + g_d') \times 13 \times 3729$	1.8배 개선
	평균 스캔 길이	6			
	평균 스캔 수	13			
공	삼각형 수	277	$(t_{rwc} + 0.5 \times 12t_{rwc}) \times 32 \times 277$	$(t_p + g_d') \times 32 \times 277$	8배 개선
	평균 스캔 길이	13			
	평균 스캔 수	32			

나 있다. 여기서, 60ns DRAM을 기준으로 할 때 DRAM의 액세스 시간 $t_p=110ns$, 읽기-수정-쓰기 사이클 시간 $t_{rwc}=155ns$, 그리고 $g_d'=20ns$ 이다. 이 경우에도 2차원 그래픽스 처리의 경우와 마찬가지로 평균 스패의 길이가 길수록 SMGRA의 성능이 좋아짐을 알 수 있고, 일반적으로 평균 스패의 길이를 10이라 가정하고 DRAM과 비교할 때 약 6배의 프레임 버퍼 액세스 속도 개선이 있음을 알 수 있다.

그림 12는 SMGRA가 3차원 연산을 처리할 때 Z값 비교기의 동작에 대해 Verilog HDL을 이용하여 시뮬레이션한 결과 파형을 보여준다.

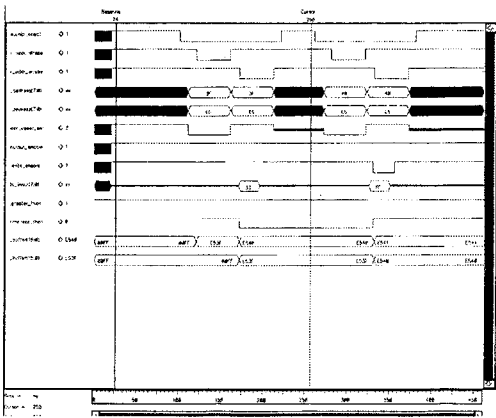


그림 12. SMGRA의 Z값 비교 연산에 대한 Verilog-XL 시뮬레이션 결과 파형

제안된 SMGRA는 C 코드로 모델링하여 기능적인 검증과 Verilog-XL을 이용하여 로직의 검증을마친 상태이고 다양한 메모리 구조에 따른 여러가지 그래픽스의 기본요소에 대해 성능분석이 진행 중이다.

4. 결론

장방형 어레이 메모리가 기본적인 DRAM의 특성인 많은 양의 메모리 용량을 가지고 임의의 모양을 가진 사각형 영역에 대한 액세스가 가능하므로 대부분의 그래픽스 응용 프로그램에서 많이 사용되는 화면 지우기와 동일한 색상값으로 사각형 영역 채우기 동작은 기존의 DRAM 구조에 비해 2차원 그래픽스 연산의 다각형 채우기의 경우는 $O(n)$, 박스 채우기의 경우는 $O(n^2)$ 의 성능개선 효과를 가지며, 선 그리기

의 경우에도 보다 빠른 실행 속도를 갖는다.

본 논문에서 제안한 그래픽스 전용 메모리는 앞서 언급한 장점을 갖는 기존의 장방형 어레이 메모리의 밴디드 주소 디코딩 로직을 개선하여 n개의 주소선을 갖는 경우 $O(\log n)$ 보다 빠른 주소 디코딩 시간 $O(1)$ 을 가지므로 메모리 사이클 시간을 줄여 줄 수 있다. 그리고 개선된 디코딩 로직은 멀티플렉싱 방법을 사용하여 필요한 주소핀 수를 $1/2$ 로 줄였으며, 하위 주소 디코딩시와 상위 주소 디코딩시에 같은 하드웨어 로직을 사용하므로 디코딩 로직을 간단하게 하였다.

또한, 그래픽스 전용 메모리는 Z값 비교기를 사용하여 3차원 그래픽스 처리시 요구되는 읽기-수정-쓰기 메모리 사이클을 단순한 쓰기 메모리 사이클로 대체하므로 프레임 버퍼 액세스 병목 현상을 약 6배 정도 개선할 수 있었다.

그래픽스 전용 메모리의 구조는 직사각형 영역에 대하여 $O(1)$ 의 주소 디코딩 시간을 갖고 있으나 입력 주소핀 수가 많은 단점이 있으므로 실제 구현시에는 메모리 구조가 갖는 성능 향상의 이점을 보장하는 범위 내에서 주소 멀티플렉싱 방법을 사용해야 할 것이다.

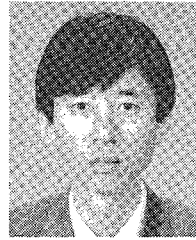
향후 연구 과제로는 3차원 그래픽스 처리시 메모리 액세스와 매우 관련이 많은 연산을 메모리 칩내에서 직접 수행하게 하여 프레임 버퍼 액세스 병목 현상을 개선하는 연구가 고려될 수 있다. 한편 최근 각광받고 있는 LEM(Logic Enhanced Memory) 기술에 부합되는 래스터라이저와 프레임 버퍼를 하나의 칩 내에 집적하여 프레임 버퍼 액세스 병목 현상을 원천적으로 제거할 수 있도록 하는 장기적인 연구도 필요하다.

참고 문헌

- [1] M. C. Whitter, "Memory Design for Raster Graphics Displays", *IEEE CG&A*, pp. 48-65, March 1984.
- [2] R. F. Sproull, "Frame Buffer Display Architectures" *Annual Review of Computer Science*, Volume 1, pp. 19-46, 1986.
- [3] Demetrescu, S., "High Speed Image Rasterization Using Scan Line Access Memories",

Proceedings of the 1985 Chapel Hill Conference on VLSI, Rockville, MD, Computer Science Press, pp. 221-243, 1985.

- [4] Molnar, S., "PixelFlow : High-Speed Rendering Using Image Composition", *Computer Graphics*, Vol. 26, No.2, pp. 231-240, July 1992.
- [5] Deering, M. F., "FBRAM: A new Form of Memory Optimized for 3D Graphics", *Proceedings of SIGGRAPH '94*, pp. 167-174, July 1994.
- [6] Whelan, D. S., "A Rectangular Area Filling Display System Architecture", *Computer Graphics*, Vol. 16, No. 3, pp. 356-362, July 1982.
- [7] 문상호, 강현석, "ARAM : 개선된 장방형 어레이 메모리", 한국정보과학회 논문지 제 23권 제 12호, 1285-1295, 1996, 12.
- [8] Bresenham, J. E., "Algorithm for Computer Control of a Digital Plotter", IBM, *Systems Journal* 4(1), pp. 25-30, 1965.
- [9] 문상호, 조준형, "프레임 버퍼 액세스 밴드width를 개선하는 두 가지 방안," 대한전자공학회 하계학술대회 논문집, 제17권 제1호, pp. 663-666, 1994, 7.
- [10] 문상호, 강현석, 박길흠, "프레임 버퍼 대역폭 개선 방안," 한국정보처리학회논문지, 제3권 제2호, pp. 407-415, 1996, 3.
- [11] Samsung Electronics Co., "CMOS Memory Data Book", 1997.
- [12] J. Foley, A. van Dam, S. K. Feiner, and J. F. Hugher, *Computer Graphics Principles and Practice*, 2nd Edition, Addison-Wesley, 1990.



김 성 진

1979년 2월 경북대학교 전자공학과 졸업(공학사)
 1981년 2월 경북대학교 대학원 전자공학과 시스템전공(공학석사)
 1995년 2월 영남대학교 대학원 전자계산기 전공(공학박사)
 1983년~1984년 삼성전자(주) 컴퓨터사업부 시스템개발실
 1996년 7월~1997년 7월 University of California at Santa Cruz, 연구교수
 1985년 3월~현재 연암공업대학 컴퓨터정보기술과 교수
 관심분야 : 컴퓨터그래픽스, 멀티미디어 통신, 초고속 정보통신망



문 상 호

1982년 2월 경북대학교 전자공학과 졸업(공학사)
 1984년 2월 경북대학교 대학원 전자공학과 졸업(공학석사)
 1998년 2월 경상대학교 대학원 전자계산학과 졸업(공학박사)
 1983년 10월~1986년 10월 삼성 전자(주) 컴퓨터사업부 시스템개발실
 1986년 11월~1995년 1월 삼성종합기술원 기반기술연구소 그래픽스팀 선임연구원
 1989년 10월~1991년 11월 미국 워싱턴 주립대학 전기 및 전자공학과 교환연구원
 1995년 3월~현재 영남이공대학 컴퓨터정보기술과 조교수
 관심분야 : 컴퓨터그래픽스, 멀티미디어, 컴퓨터구조