
모바일 그래픽스 응용을 위한 부동소수점 승산기의 설계

최병윤* · Zoran Salcic**

Design of Floating-Point Multiplier for Mobile Graphics Application

Byeong-yoon Choi* · Zoran Salcic**

이 논문은 동의대학교 2006년도 교내 연구 과제와 정통부 선도 기반 과제에 의한 결과임

요 약

본 논문에서는 2단 파이프라인 구조의 부동 소수점 승산기 회로를 설계하였다. 부동 소수점 승산기는 3차원 그래픽 API인 OpenGL과 Direct3D를 위한 단일 정밀도 곱셈 연산을 지원하며, 포화 연산, 면적 효율적인 점착(sticky) 비트 발생기 및 플래그 프리픽스 가산기를 결합하여, 면적 효율적이며 적은 파이프라인 지연 구조를 갖는다. 설계된 회로는 0.13 μ m CMOS 표준 셀을 사용하여 합성한 결과 약 4-ns의 지연시간을 갖고 있으며, 약 7,500개로 구성된다. 설계된 부동 소수점 승산기의 최대 연산 성능은 약 250 MFLOPS이므로, 3차원 모바일 그래픽 분야에 효율적으로 적용 가능하다.

ABSTRACT

In this paper, two-stage pipelined floating-point multiplier (FP-MUL) is designed. The FP-MUL processor supports single precision multiplication for 3D graphic APIs, such as OpenGL and Direct3D and has area-efficient and low-latency architecture via saturated arithmetic, area-efficient sticky-bit generator, and flagged prefix adder. The FP-MUL has about 4-ns delay time under 0.13 μ m CMOS standard cell library and consists of about 7,500 gates. Because its maximum performance is about 250 MFLOPS, it can be applicable to mobile 3D graphics application.

키워드

부동 소수점 승산기, 3차원 그래픽스, 모바일 디바이스, OpenGL-ES, Direct3D, 셰이더

I. 서 론

3차원 컴퓨터 그래픽스 기술의 발달로 이제 사용자들은 PC나 노트북을 통해 딱딱한 2차원 영상이 아닌 살아 있는 3차원 영상을 볼 수 있게 되었고, 나아가 PDA와 같은 휴대용 단말기에서도 3차원 영상을 볼 수 있을 것이

다[1]. 그런데 PDA, Mobile Phone에서 3차원 게임을 구현하기 위해 고속 그래픽 가속기가 필요한데, PC환경과 달리 휴대용 단말기에서는 면적 조건과 저전력 조건이 엄격하므로 이러한 환경에 적합한 그래픽 가속기는 다른 특성의 하드웨어 구조를 필요로 한다[2]. 그리고 그래픽 프로세서에서 응용 분야에 적합한 다양한 알고리즘을

* 동의대학교 컴퓨터공학과

** University of Auckland University, New Zealand

구현하고, 시장 환경에 융통성 있게 대처하기 위해, 기존의 고정된 연산 하드웨어가 아닌 프로그램 가능한 정점 셰이더와 픽셀 셰이더 연구가 필요하다[3]. 셰이더 프로세서에 필요한 핵심 부품은 부동 소수점 연산장치(FP-AU), 부동소수점 곱셈기(FP-MUL) 과 초월 함수를 처리하는 특수 기능 연산장치이다. 본 논문에서는 모바일 그래픽 분야에 적용을 위한 면적 효율적이고, 낮은 파워라인 지연 특성을 갖는 부동 소수점 승산기 회로를 설계하고 성능을 분석하였다.

본 논문의 구성은 다음과 같다. 제 2장에서는 3차원 그래픽 가속기 구조와 부동 소수점 승산기의 설계 사양을 살펴보고, 제 3 장에서는 부동 소수점 승산기의 하드웨어 설계를 기술하였으며, 4장에서는 설계한 회로에 대한 검증과 성능 분석을 하였으며 마지막으로 결론을 기술하였다.

II. 모바일용 3차원 그래픽 프로세서 구조와 설계 사양

본 장에서는 3차원 그래픽 가속기 구조와 부동 소수점 승산기의 설계 사양을 기술하였다.

2.1 셰이더 기반 그래픽 가속기 구조

최근 기존 3차원 그래픽 가속기의 고정 파이프라인 처리의 융통성 제약문제를 해결하기 위해, 프로그램 가능한 프로세서 개념의 셰이더 기반 구조가 데스크탑 PC에서부터 모바일 디바이스로 응용이 확대되고 있다. 셰이더 구조의 그래픽 가속기는 정점 셰이더, 픽셀 셰이더와 래스터라이저 회로로 구성된다. 본 연구에서 개발 중인 모바일용 3차원 그래픽 가속기용 정점 셰이더의 구조와 명령어의 파이프라인 구조는 각각 그림 1, 그림 2와 같다. 정점 셰이더는 4개의 연산 장치가 병렬로 수행되는 SIMD 형태의 구조로 되어있다.

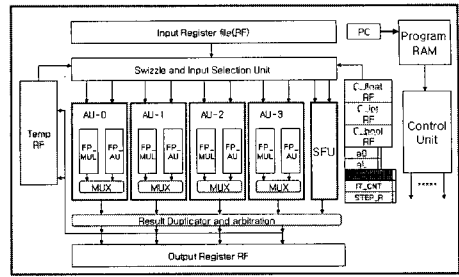


그림 1. 정점 셰이더 구조
Fig. 1 Block diagram of vertex shader

정점 셰이더의 경우 6단계 명령어 파이프라인은 IF, AG, OF, EX1, EX2, WB 단계로 구성된다. 그림 2의 명령어 파이프라인 사양에 따라 2개의 파이프라인 단계(EX1, EX2)가 부동 소수점 승산기(FP-MUL)에 할당될 수 있다. 높은 동작 주파수를 위해 FP-MUL을 많은 파이프라인 단계로 구현할 수 있지만, 많은 파이프라인 단계는 셰이더 명령어간 데이터 의존 관계를 증가시켜 성능을 저하시킬 수 있다. 따라서 본 연구의 부동 소수점 승산기의 경우 2단계의 파이프라인 방식을 갖도록 설계하였다.

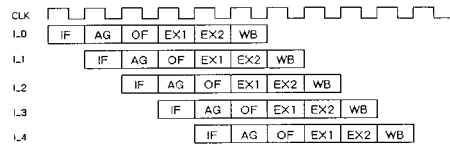


그림 2. 정점 셰이더 명령어의 파이프라인
Fig. 2 Pipeline of vertex shader instruction

2.2 부동 소수점 승산기의 설계 사양

그래픽 처리의 설계 기준을 바탕으로 본 연구의 부동 소수점 승산기는 기존 마이크로프로세서의 부동 소수점 승산기와 다른 설계 사양을 갖고 있다.

첫째, 기존 컴퓨터의 부동 소수점 승산기와 달리 본 연구의 그래픽용 FP-MUL의 경우 그림 3과 같이 32-비트 단정도 데이터 형식만을 지원한다.

둘째, 그래픽 분야의 경우 IEEE 표준안 만족보다는 속도와 정밀도 허용 한도 내의 연산 정확성이 중요하므로, 본 승산기는 Round-to-Nearest Even(RNE) 반올림 모드만 지원한다.

셋째, 본 FP-MUL의 경우 예외 조건을 감지하는 경우 예외 처리 루틴을 호출하는 기법 대신에 포화 연산을 적용하여 이러한 문제 발생을 해결하였다.

넷째, 지수의 범위 확대를 위해 최대 지수 값, 255를 무한대와 NaN으로 사용하지 않고, 실제 지수 값으로 사용하여 지수 범위를 확대하도록 하였다.

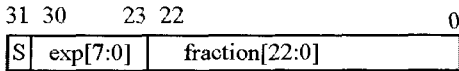


그림 3. 단정도 부동소수점 데이터 형식
Fig. 3. Single-precision floating-point data format

III. 승산기의 하드웨어 설계

FP-MUL 회로는 설계 사양에 따라 2단 파이프라인 구조로 구현되었으며, 크게 부호 처리부, 지수 처리부, 소수 처리부로 나누어진다.

3.1 소수 처리부의 설계

소수 처리부는 2단 파이프라인 구조로 구현되었는데, 첫 번째 파이프라인 단계는 Booth 부호화, 부분곱 생성 회로, Wallace 트리 단계로 구성되며, 단계 2에서 필요한 합 벡터와 캐리 벡터를 생성한다. 두 번째 단계는 단계 1에서 생성한 합과 캐리 벡터를 사용하여 반올림, 정규화 및 재정규화 알고리즘을 적용하여 최종 결과의 소수부 값을 결정한다. 본 승산기에 사용하는 곱셈 알고리즘은 수정된 Booth 승산 알고리즘을 사용하였다[5]. 단계 1의 Booth 리코딩 과정은 소수부의 기존 23-비트, 감춤 비트와 부호 비트(항상 0)가 포함되어 그림 4와 같이 수행된다. 부분 곱은 2의 보수 형식을 갖고 있기 때문에, 여러 개의 부분 곱을 더할 경우 많은 면적이 필요한 부호 확장이 필요하다.

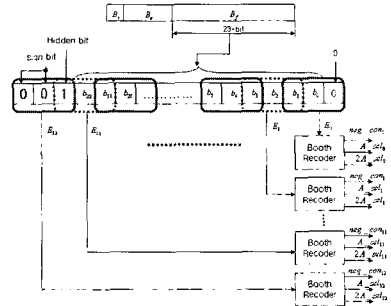


그림 4. Booth 리코딩 동작
Fig. 4. Booth recoding operation

본 연구에서는 이러한 문제를 해결하기 위해 참고 문헌[6]에서 제안한 기법을 수정하여 사용하였다. 일반적인 부호 확장 제거 기법의 경우 PP[0]의 최상위 비트, P[msb] 위치에 1이 함께 놓여야 하는 문제가 있어서, 부분곱 PP[0] 부분에 적용되는 부호 확장 기법을 식(1)과 같이 수정하였다.

$$11 + P[msb] = P[msb] P[msb] P[msb] \quad (1)$$

그림 5는 본 승산기에 사용한 부분곱 생성 방법과 부호 확장 제거 기법을 나타낸다. PP[0]을 제외한 나머지 부분곱 PP[i]는 2-비트만큼의 부호 확장 처리가 필요하다. 생성된 13개의 부분 곱은 Wallace 트리를 통해 48-비트의 합 벡터와 캐리 벡터로 변환된다. 그림 6은 본 승산기에 사용된 Wallace 트리 회로를 나타낸다.

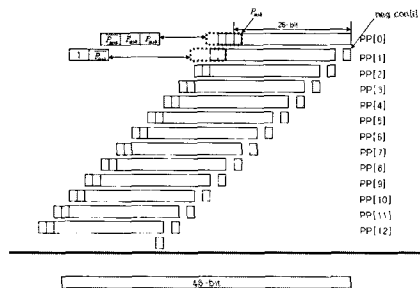


그림 5. 부분곱 생성 및 부호 확장 제거
Fig. 5. Partial product generation and sign extension elimination

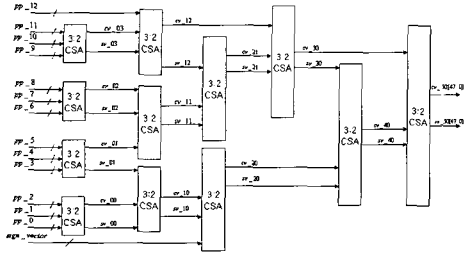


그림 6. Wallace 트리
Fig. 6. Wallace tree

본 연구의 Wallace 트리[7]의 경우 전가산기 구조의 (3,2) 컴프레서를 사용하여 구현되었다. 그림 6에서 부호_벡터부분은 그림 5에서 $neg_con[i]$ 에 의해 2의 보수 값 구현을 위해 부분곱의 최하위 비트에 추가되는 1을 묶어서 그림 7과 같이 정의된 값이다. 이와 같이 할 경우 $neg_con[i]$ 에 의한 추가의 1의 덧셈 동작이 하나의 부분 곱과 같이 처리될 수 있어서 Wallace 트리에 쉽게 적용이 가능하다. 24-비트 × 24-비트 부동 소수점 곱셈 동작을 분석해 보면, 소수부 결과 값이 정규화된 형식에 비해 최대 1-비트만큼 오버플로우될 수 있다[8]. 따라서 소수부 오버플로우 여부에 따라 반올림 위치가 다르게 된다.

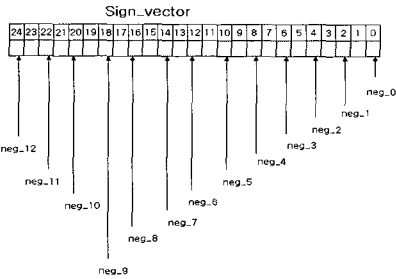


그림 7. 부호_벡터 생성 기법
Fig. 7. sign_vector generation scheme

본 승산기에서는 단계 2에서 소수부 덧셈과 반올림, 정규화 과정을 통합하기 위해서 그림 9의 구조를 채택하였다[9]. 병렬 반올림 동작을 하려면, 그림 9의 복합 가산기로 구현되는 상위 부분(bit[47] ~ bit[25])에 더해지는 캐리 입력 값을, 반올림 여부에 무관하게 0 또는 1로 제한할 수 있는 구조가 필요하다. 이를 위해 캐리 보존 동작을 하는 반가산기 계층을 두었다. 그 경우 반가산기 계층을 통과한 비트[24:22]의 값의 범위는 그림 10의 [case 2]에 보이는 바와 같이 10이하로 제한된다. 따라서

하위부의 carry_in(C₂₂)조건과 반올림 신호가 함께 입력되더라도, 상위부, 즉 복합가산기로 최대 1개의 캐리 출력만 발생한다. 따라서 상위부를 A+B, A+B+1의 2개의 출력을 생성하는 복합 가산기(compound adder)로 구현할 수 있다.

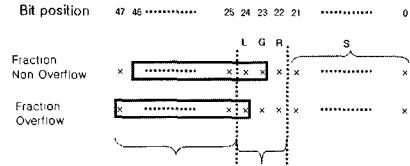


그림 8. 곱셈 결과 값과 반올림 관련 비트
Fig. 8. Multiplier's result and rounding-related bits

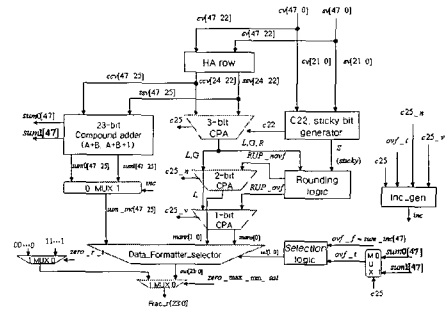


그림 9. 반올림과 정규화 동작을 병렬로 처리하는 구조

Fig. 9 Structure that executes rounding and normalization in parallel

그림 10의 동작 특성을 활용하면, 그림 9와 같이 48 비트 입력 데이터를 3개의 부분으로 나누어 병렬 반올림, 정규화, 재정규화 처리가 가능하다. 순차적인 덧셈, 반올림, 정규화, 재정규화 동작을 병렬로 수행하려

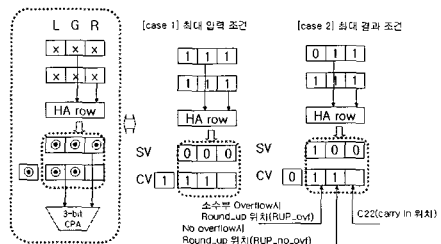


그림 10. L, G, R 비트 위치의 반가산기 행의 동작
Fig. 10. Operation of HA row for L, G, R bit positions

면, 예상 결과를 미리 병렬로 발생해두었다가 결정된 조건에 따라 적절하게 선택해야 한다. 그러나 내부적인 순차적인 특성을 지원하기 위해 먼저 합 벡터와 캐리 벡터의 덧셈을 통해 반올림전 소수부 오버플로우(ovf_t)를 결정해야 한다. 이 값은 복합 가산기의 A+B, A+B+1의 출력을 sum0, sum1이라 할 경우, 하위부에서 발생한 캐리 입력 값(C25)에 의해 최상위 비트 값 sum0[47]과 sum1[47]중 바람직한 값을 선택함에 의해 결정된다. 결정된 ovf_t에 따라 반올림 위치와 반올림 신호(RUP_no_ovf, RUP_ovf)가 결정된다. 그림 9에서 2-비트 CPA(carry propagate adder)은 소수부 오버플로우가 발생치 않은 경우, 최종 결과 값의 하위 2 비트(mann[1:0])와 반올림 결과를 반영한 복합 가산기의 최종 출력을 선택하기 위한 C25_n 신호를 생성한다. 반면 1-비트 CPA는 소수부 오버플로우가 발생한 경우 결과 값의 하위 1 비트(manv[0])와 반올림 결과를 반영한 복합 가산기의 최종 출력을 선택하기 위한 C25_v 신호를 생성한다. 반올림을 반영한 최종 복합 가산기에 대한 선택 신호 inc는 식(2)과 같이 정의된다.

$$inc = C_{25} + C_{25_v} \cdot ovf_t + C_{25_n} \cdot (ovf_t) \quad (2)$$

단, 반올림 전에는 소수부 오버플로우가 발생치 않았지만, 반올림 동작에 의해 최종적인 오버플로우(ovf_f)가 발생하는 경우가 가능하다. 즉 inc에 의해 선택한 결과 sum_inc[47:25]의 최상위 비트 sum_inc[47]이 반올림을 반영한 최종 오버플로우(ovf_f)가 된다. 위의 2가지 오버플로우 값(ovf_f, ovf_t)을 사용하여 반올림, 정규화, 재정규화가 반영된 최종 결과 선택 방법이 표 1과 같이 정의된다.

표 1. 최종 결과 선택 방법
Table 1. Final result selection scheme

ovf_f	ovf_t	최종 결과
0	0	sum_inc[46:25], mann[1:0]
1	0	sum_inc[47:25], mann[1]
-	1	sum_inc[47:25], manv[0]

그림 9에서 C₂₂, 점착 비트 발생기는 3-bit CPA에 입력되는 캐리 입력 값과 RNE 반올림 동작 모드에 필요한 점착 비트를 생성하는 회로이다. 입력 데이터의 하위부 0

의 개수의 합을 이용한 기존 점착 회로는 많은 면적이 필요한 단점이 있다[8,10].

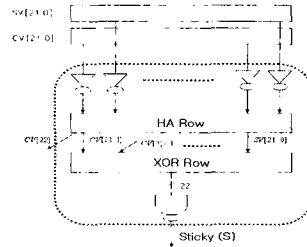


그림 11. 점착 비트 발생 회로
Fig. 11. Sticky bit generator

따라서 본 승산기에서는 참고 문헌 [11]에서 제안한 A+B=K 조건을 고속으로 감지하는 회로를 변형하여, A+B ≠ 0을 고속으로 결정하는 방식에 의해 그림 11과 새로운 점착 비트 발생회로를 설계하였다. 설계된 점착 비트 회로는 기존 방식에 비해 약 30% 이상의 면적 감소가 가능하다. 그림 9에서 A+B, A+B+1을 병렬로 계산하는 복합 가산기는 일반적으로 캐리 선택 가산기를 변형한 2개의 가산기로 구현되는데, 이 방식은 많은 면적이 필요하다. 본 승산기에서는 지연 증가(late increment) 기능을 갖춘 플래그 프리픽스 가산기(flagged prefix adder)[14]를 수정해서 구현하였다. 본 승산기에 필요한 복합 가산기는 반올림을 수행하기 전 오버플로우 발생 여부를 감지하는 비트(ovf_t)를 결정하기 위해, A+B, A+B+1의 최상위 비트(sum0[msb], sum1[msb])를 결정하는 SG0와 SG1 회로를 갖고 있다. 그림 9의 복합 가산기 부분을 플래그 프리픽스 가산기로 대체함에 의해서 기

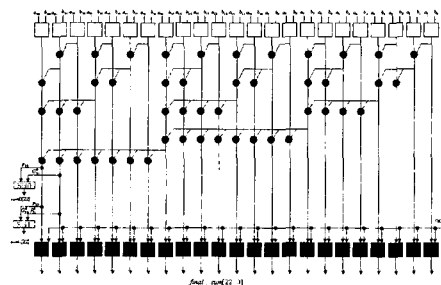


그림 12. sum0[22], sum1[22]출력을 갖는 플래그 프리픽스 가산기
Fig. 12. Flagged prefix adder with sum0[22] and sum1[22] outputs

존 캐리 선택 가산기를 사용하는 복합 가산기에 비해 약 20%의 게이트 수 감소를 얻을 수 있었다.

3.2 지수 처리부의 설계

IEEE 754 부동 소수점 데이터의 경우 지수부에 127의 바이어스(bias)값이 포함되어 있기 때문에 지수부 연산을 수행하려면, 식 (3)의 동작이 필요하다.

$$EXP_r = EXP_A + EXP_B - bias \quad (3)$$

식(5)의 동작은 2개의 덧셈 동작이 직렬로 연결되는 특성이 있다. 이러한 문제를 해결하기 위해, 본 승산기에서는 그림 13과 같이 캐리 보존 덧셈기와 캐리 전달 덧셈기를 사용하여 해결하였다.

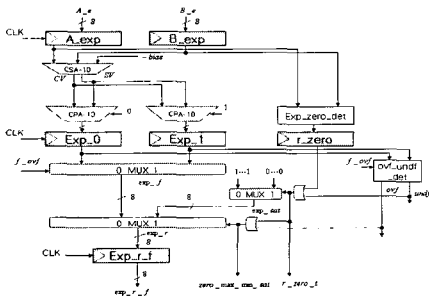


그림 13. 지수부 회로의 블록도
Fig. 13. Block diagram of exponent circuit

단, 소수부 곱셈 과정에 최대 1-비트 만큼의 소수부 오버플로우(ovf_f)가 발생할 수 있으므로, 지수부의 값이 1 만큼 추가로 증가될 수 있다. 따라서 파이프라인 단계 2에서 지수부 오버플로우와 언더플로우, 최종 지수부 값 계산을 병렬로 하기 위해 지수부 파이프라인 단계 1의 CPA는 캐리 입력을 0과 1로 하는 2가지 가산기로 구성된다. 파이프라인 단계 2에서는 지수부 오버플로우와 언더플로우 조건을 병렬로 계산하여, 포화 연산에 따라서 결과 값을 레지스터 표현 최대 값 또는 0으로 설정한다.

IV. 설계 검증 및 성능 분석

부동 소수점 승산기를 설계하기 위해 FP-MUL에 대한 테스트 벡터와 예상 검증 결과를 생성하는 C-프로그램

모델을 개발하였다. 이러한 C-모델에서 얻어진 값은 Verilog-HDL로 설계한 회로에서 얻어진 결과와의 비교를 통해 올바른 결과 검증에 활용되었다. 그림 14와 15는 복합 가산기와 전체 FP-MUL에 대한 Modelsim 검증 파형을 나타낸다. 설계된 회로는 Modelsim 검증을 마친 후, 0.13 μ m CMOS 표준 셀 라이브러리와 Synopsys 합성 소프트웨어를 사용하여 합성한 결과 약 7,500개의 게이트로 구성되며 최장 전달 지연 시간은 약 4-ns로 최대 동작 주파수는 약 250 Mhz를 가짐을 알 수 있었다. 표 2는 설계된 FP-MUL의 전기적 특성을 나타낸다. 그리고 표 3은 설계된 회로와의 기존 회로와의 비교 결과를 나타낸다.

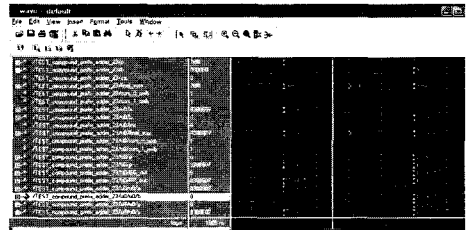


그림 14. 복합 가산기에 대한 모의 검증
Fig. 14. Simulation of compound adder

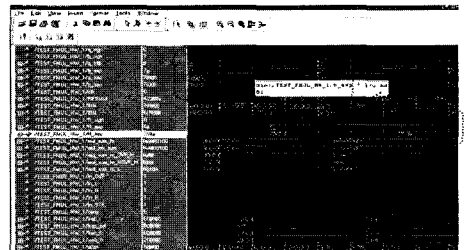


그림 15. FP-MUL에 대한 모의 검증
Fig. 15. Simulation waveform of FP-MUL

표 2. 전기적 특성
Table 2. Electric characteristics

공정	0.13 μ m standard cell
파이프라인 수	2
반올림 방식	round-to-nearest even
데이터 형식	IEEE 단정도 형식
오버플로우 처리	최대 값으로 포화 처리
복합가산기 구조	수정된 flagged prefix adder
게이트 수	약 7,500
동작 주파수	250 Mhz
최대 연산 성능	250 MFLOPS

표 3. 성능 비교
Table 3. Performance comparisons

특성	방식	참고문헌 [9]	참고문헌 [13]	참고문헌 [14]	본 연구 승산기
동작 주파수		167 Mhz	240 Mhz	600 Mhz	250 Mhz
latency		3	2	4	2
지원 데이터		단정도, 배정도 MUL	54×54 -비트 MUL	32-bit MAC	단정도 MUL
포화연산		미지원	미지원	미지원	지원
반올림 방식		모든 모드	-	-	RNE 모드
게이트수		-	15,000	-	7,500
공정 기술		0.5μm CMOS	0.25μm CMOS	0.18μm CMOS	0.13μm CMOS
응용 분야		범용 FPU	그래픽 응용	DSP 응용	그래픽 응용

본 연구에서 설계한 부동 소수점 승산기는 기존 승산기와 비교해 볼 때, 파이프라인 구조를 갖고 있으며, 3차원 그래픽 연산에 필수적인 포화연산을 지원하고 있으며 약 250 MFLOPS의 성능을 갖고 있으므로, 모바일 그래픽 응용에 효율적으로 적용 가능하다고 판단된다. 최근 SIMD 구조의 그래픽 프로세서는 사용 빈도가 높은 부동 소수점 데이터 형태의 MAC 연산의 성능을 향상시키기 위해, 곱셈 연산과 덧셈 연산을 통합하는 구조가 많이 제안되고 있다. 따라서 본 연구 결과의 성능 개선을 위해 MAC 구조에 대한 연구가 필요하다고 판단된다.

V. 결론

본 논문에서 설계한 FP-MUL 회로는 본 연구의 최종 목표인 정점 셰이더 회로의 설계 사양에 맞추어 데이터 의존 지연을 최소화할 수 있도록 2단 파이프라인 구조와 포화 연산 사양을 만족한다. 반올림, 정규화, 재정규화를 파이프라인 단계 2에서 하나의 클럭 사이클 내에 처리하기 위해, 병렬 반올림, 정규화 기법을 사용하였다. 그리고 면적 효율적인 새로운 점착 비트 발생 회로와 수정된 플래그 프리픽스 가산기를 바탕으로 한 새로운 복합 가산기를 사용하여 기존 방식에 의해 면적 특성을

약 20% 이상 개선할 수 있었다. 그리고 포화 연산을 사용함에 의해 3차원 그래픽 프로세서에서 요구되는 중단이 없는 실시간 처리가 가능하다. 설계된 회로는 0.13μm CMOS 표준 셀 공정 조건에서 약 7,500개의 게이트로 구성되며, 250 MHz의 동작 주파수로 최대 250 MFLOPS의 연산 성능을 갖고 있어서, 모바일 그래픽스 회로와 셰이더 회로에 효율적으로 적용 가능하다고 판단된다.

감사의 글

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

참고문헌

- [1] 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.
- [2] Dave Astle and Dave Durnil, Opicro OpenGL ES Game Development, PTR, 2004.
- [3] Kris Gray, Microsoft DirectX9 Programmable Pipeline, Microsoft Press 2004.
- [4] IEEE, ANSI/IEEE Standard 754-1985: IEEE Standard for Binary Floating-Point Arithmetic, IEEE Press, 1985.
- [5] Rubinfeld, L.P, "A proof of the modified Booth's algorithm for multiplication," IEEE Transaction on Computer, vol.24, no.10, pp.1010-1015. October 1975.
- [6] M. Rooda, "Method to reduce the sign bit extension in a multiplier that uses the modified booth algorithm," Electronics Letters, vol 22, no.20, pp.1014-1015. 25h September, 1986.
- [7] C. S. Wallace, "A suggestion for parallel multipliers," IEEE Trans. Electron. Computer, no. EC-13, pp.14-17, Feb, 1964.
- [8] Mark R. Santoro, Gary Bewick and Mark A. Horowitz, "Rounding Algorithms for IEEE Multipliers", IEEE 9th Symposium on Computer Arithmetic, pp.176-183. 1989.

- [9] Robert K. Yu and Gregory B. Zyner, "167 MHz Radix-4 Floating Point Multiplier", IEEE 12th Symposium on Computer Arithmetic, 1 pp.149-154. 1995.
- [10] V.G. Oklobdjija, etal, "An algorithm and novel design of leading zero detector: comparison with logic synthesis," IEEE Transactions on VLSI, vol.2, pp.124-128, March 1994.
- [11] J. Cortadella and J. M. Liaberia, "Evaluation of A+B=K conditions without carry propagation", IEEE Transaction on Computers, vol. 41, no.11, pp.1484-1488, Nov. 1992.
- [12] Neil Burgess, "The flagged prefix adder and its applications in integer Arithmetic," Journal of VLSI Signal Processing, vol 31, pp.263-271, 2002.
- [13] G. Goto, A. Inoue, and T. Izawa, "A 4.1-ns Compact 54 × 54-b Multiplier Utilizing Sign-select Booth Encoders," IEEE JSSC, vol.32, no.11, pp.1676-1682, November, 1997.
- [14] Yuyun Liao and David B. Roberts, "A High-Performance and Low-Power 32-bit Multiply-Accumulator Unit with Single-Instruction-Multiple-Data (SIMD) Feature," IEEE JSSC, vol.37, no.7, pp.926-931, August, 2004.

저자소개



최 병 윤(Byeong-Yoon Choi)

1985년 2월 연세대학교 전자공학과 졸업

1992년 8월 연세대학교 전자공학과 공학 박사

2006년 1월 ~ 12월 뉴질랜드 오클랜드 대학 방문 연구 교수

1993년 3월 ~ 현재 동의대학교 교수

※관심분야: RISC와 SoC 설계



Zoran Salcic

현재: 뉴질랜드 University of Auckland 교수

※관심분야: 임베디드 시스템 설계