

멀티미디어 처리에 적합한 SIMD 곱셈누적 연산기의 설계

홍인표, 정재원, 정우경, 이용석
연세대학교 전기전자공학과 프로세서 연구실
전화 : 02-2123-2872

SIMD Multiply-accumulate Unit Design for Multimedia Data Processing

In Pyo Hong, Woo Kyeong Jeong, Jae Won Jeong, Yong Surk Lee
Dept. of Electrical and Electronic Engineering, Yonsei University
E-mail : necross@dubiki.yonsei.ac.kr

Abstract

In this paper, a SIMD 64bit MAC (Multiply-ACcumulate) unit is designed. It is composed of two 32bit MAC unit which supports SIMD 16bit operations. As a result, It can process two 32bit MAC operations or four 16bit operations in one cycle.

Proposed MAC unit is described in Verilog HDL. After functional verification is performed, MAC unit is synthesized and optimized with 0.35 μ m standard cell library. The synthesis result shows that this MAC unit can operate at 80MHz of clock frequency in 85 $^{\circ}$ C, 3.0V, worst case process and 125MHz of clock frequency at 25 $^{\circ}$ C, 3.3V, typical case process. It achieves 320Mops of performance, and is suitable for embedded DSP processors.

I. 서론

영상, 음성 등의 멀티미디어 자료 처리에 대한 수요가 늘어나면서, 최신의 고성능 범용 마이크로 프로세서들은 이에 대응하기 위해 64~128비트 데이터 폭을 갖는 SIMD(Single Instruction Multiple Data) 연산기를 내장하고 있다. 하지만, 이런 연산기들은 천만개 정도의 트랜지스터를 집적한 이들 프로세서에서 상당 부

분의 면적을 차지하고 있으며, 다양한 명령어 지원으로 구조가 복잡하고 설계가 까다롭다. 이러한 연산기를 내장형 혹은 DSP 프로세서에 내장하는 것은 적합하지 않으므로, 고성능의 SIMD 연산기들에 비해서 작은 면적을 차지하고, 간단한 핵심 동작만을 하는 SIMD 연산기에 대한 연구가 필요하다.

본 논문에서는 64비트의 데이터 폭을 갖는 SIMD MAC 연산기를 설계하였다. 본 연산기는 한 사이클에 네 개의 16x16+64비트 또는 2개의 32x32+64비트 MAC 연산을 할 수 있으며, 32비트 데이터에 대하여는 signed/unsigned 연산을 지원한다. 연산기는 HDL로 설계하여 검증하였으며, 0.35 μ m 표준 셀로 합성하여 최악조건에서 80MHz의 동작 주파수를 보인다.

II. SIMD MAC 연산기의 구조

2.1 SIMD MAC 연산기의 기본 구조

64비트의 SIMD MAC 연산기는 두 개의 32비트 SIMD MAC 연산기를 내장하고 있다. 그림2-1에서 보는 것과 같이 두 개의 32비트 SIMD 연산기가 병렬로 연결되어 있으며, 하나의 32비트 연산기는 32비트 곱셈기와 2개의 64비트 누적 저장기(accumulator register)로 구성되어 있다. 곱셈기에는 누적 연산을 위

한 누적 덧셈기가 포함되어 있다.

그림 2-1. SIMD MAC 연산기의 기본구조

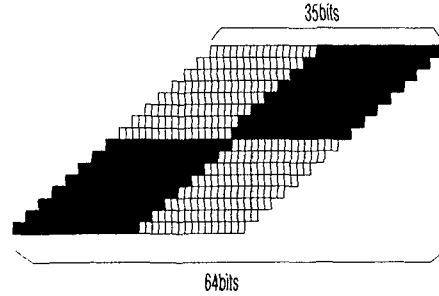
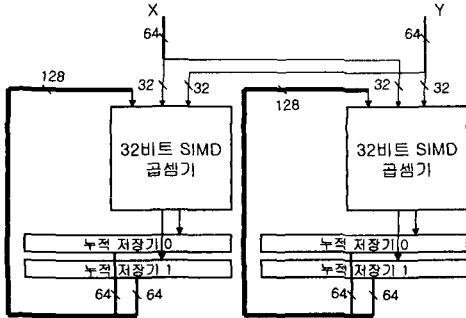


그림 2-3. 부분곱의 분할

2.2 32비트 SIMD MAC Block의 세부 구조

32비트 SIMD MAC은 하나의 32비트 MAC 연산 혹은 두 개의 16비트 MAC 연산을 한 사이클에 처리할 수 있으며, 32비트의 radix-4 수정 Booth 알고리즘을 사용한 곱셈기와 윌레스 트리에 통합된 누적 덧셈기로 구성되어 있다. 그림 2-2에서 전체적인 구성을 볼 수 있다.

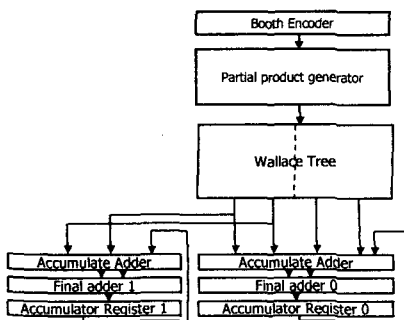


그림 2-2. 32비트 SIMD MAC의 구조

32비트 곱셈기를 두 부분으로 나누어 각 부분이 독립적으로 16비트 연산을 할 수 있도록 구성하였고, 누적 덧셈기와 최종 덧셈기(final adder)는 두 개의 16비트 데이터를 처리할 수 있도록 두 개씩을 내장하였다.

Booth encoder는 32비트를 16비트씩 쪼개어 Booth 알고리즘을 적용할 수 있도록 설계하였고, 부분곱 생성기(partial product generator)는 동작 모드에 따라 19비트 혹은 35비트의 부분곱을 만들 수 있도록 설계하였다. 그림 2-3에서는 16비트 SIMD 연산시에 부분곱들을 생성하는 것을 보여준다. 흰색으로 나타난 부분은 16비트 연산시에는 사용하지 않는 영역이다.

32비트 연산인 경우는 32비트의 피승수에 부호생성 알고리즘에 의한 3비트를 추가하여 35비트의 부분곱이 되며, 16비트 연산인 경우는 19비트가 된다. 16비트 연산시에 각 19비트 부분곱은 그림 2-3와 같이 윌레스 트리의 적절한 위치로 입력되게 되며, 그림 2-4와 같이 윌레스 트리의 32번째 비트열에서 33번째 비트열 사이의 자리올림(carry propagation)을 차단하여 좁으로써 두 개의 16비트 연산이 독립적으로 이루어질 수 있도록 하였다.

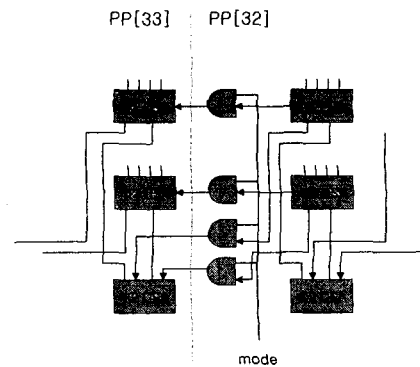
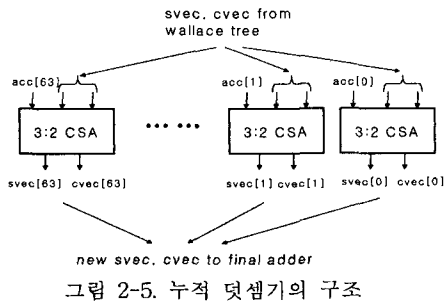


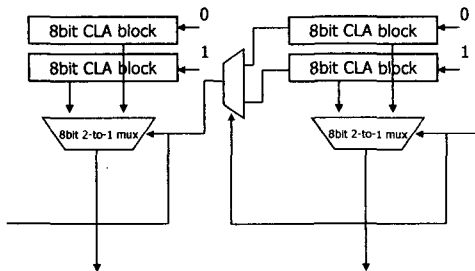
그림 2-4. 윌레스 트리의 자리올림 조절

윌레스 트리에서 부분곱들의 덧셈으로 나온 합벡터와 자리올림벡터는 최종 덧셈기로 들어가기 전에 누적 덧셈하는 과정을 거친다. 최종 덧셈기에서 곱셈 결과를 확정된 후 누적 덧셈을 하려면 누적 덧셈을 하는 과정에서 64비트의 자리올림이 다시 발생하므로, 지연시간이 증가하여, 한 사이클에 MAC연산을 마칠 수 없게 된다. 따라서 누적 저장기에 저장된 누적 값을 윌레스 트리에서 하나의 부분곱과 같이 취급하여 다른 부분곱들과 같이 더하여주면, 자리올림으로 인한 지연시간의 증가를 줄일 수 있다. 윌레스 트리는 덧셈 수행시간과 자리올림이 전파되는 시간을 중첩시켜 자리올림에 의한 지연시간의 증가를 줄일 수 있도록 설계된 구조이므로, 누적값을 윌레스 트리에서 입력받아 처리하도록 하면 하나의 3:2

CSA의 지연시간 만으로 64비트 누적 덧셈을 한 것과 같은 결과를 얻을 수 있다. 16x16비트의 곱셈결과를 64비트의 누적 저장기에 더하기 위해서는 부호 확장이 필요하므로 누적 덧셈기의 지연시간은 부호확장을 위한 AND 게이트와 3:2 CSA의 지연시간을 합친 것이 된다.



최종 덧셈기는 64비트 덧셈기로, 월레스 트리와 누적 덧셈을 거쳐 나온 64비트의 합벡터와 자리올림벡터를 더하는 역할을 한다. 최종 덧셈기는 누적 덧셈기와 같이 자리올림 전파를 없앨 수 없으므로, 고속의 덧셈기를 사용하여야 한다. 본 SIMD MAC 연산기에서는 4비트, 8비트, 16비트, 32비트 단위의 덧셈기를 CLA(Carry Lookahead adder)구조로 설계하고, 임계 경로를 분석하여 이들 각 CLA 블록들을 적절한 위치에 배치하여 carry select 구조로 연결함으로써 최종 덧셈기를 구현하였다. Carry select 구조는 64비트의 큰 덧셈기를 작은 단위로 나누어 자리올림이 있는 경우와 없는 경우를 모두 연산한 후 자리올림 입력에 따라 결과를 선택하는 구조로서, 곱셈기를 잘게 쪼갤수록 덧셈에 걸리는 시간은 줄어들지만, 결과를 선택해야 할 MUX가 많아지므로, 이들간의 상호관계와 임계경로 분석을 통하여 최적의 분할을 했다.



III. SIMD MAC 연산기의 설계 및 검증

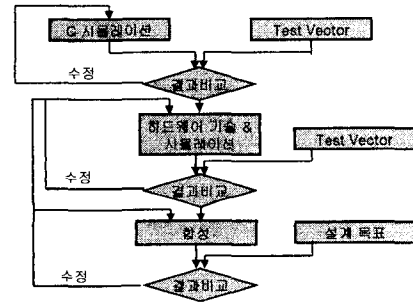
3.1 SIMD MAC 연산기의 설계 및 검증 과정

SIMD MAC 연산기는 기존의 32비트 곱셈기의 구조를 수정하고, 약간의 조절신호들을 추가하여 설계한 것으로, 하드웨어 설계를 하기 전에 자세한 구조에 대한 시뮬레이션이 필요하였다. 따라서 본 연산기의 목적을 만족할 수 있는 구조를 제안한 후, 이를 C 언어를 이용한 시뮬레이션 프로그램을 작성하여 제안된 구조가 제 기능을 수행할 수 있는 가를 확인하였다. 또한 시뮬레이션을 통하여 조절신호들이 인가되어야 할 위치와 SIMD 연산을 위한 자리올림 조절등의 세부적인 사항을 확정하였다.

구조에 대한 시뮬레이션이 끝난 후, 하드웨어를 Verilog HDL로 기술하였다. HDL은 gate level로 작성하였고, 작성된 Verilog Code를 이용하여 기능 검증을 수행하였다. 기능 검증은 SIMD 16비트 연산, signed 32비트 연산, unsigned 32비트 연산 각각에 대하여 10만개의 random test vector를 발생시켜 Sparc™에서 수행한 결과와 본 연산기의 HDL 시뮬레이션 결과를 비교함으로써 이루어졌다.

기능 검증을 마친 하드웨어는 Synopsys Design Analyzer™를 이용하여, 0.35µm 표준 셀 라이브러리로 합성하였다. 합성 결과를 확인하여 설계 목적에 미달하는 경우에는 임계경로 분석을 통하여 하드웨어의 세부 구조를 변경하거나 합성 방법을 바꾸어 설계 목적에 더욱 근접하도록 하였다.

이 과정을 design flow로 나타내면 그림3-1과 같다.



3.2 SIMD MAC 연산기 합성 결과

합성된 SIMD MAC 연산기는 85°C, 3.0V, 최악 공정 조건에서 80MHz의 동작 주파수를 보이고, 25°C, 3.3V, 표준 공정 조건에서는 125MHz의 동작 주파수를 가진다. 회로의 각 임계경로와 그에 따른 지연시간을 그림3-2에 표시하였다.

VI. 결론

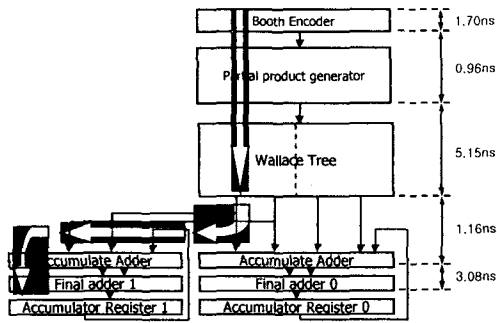


그림 3-2 임계경로와 지연시간

본 연산기는 64비트의 데이터 폭으로 한 사이클에 최대 4개의 16비트 데이터를 MAC 연산할 수 있으므로, 80MHz로 동작시키는 경우, 초당 320Mops의 성능을 보인다. 본 연산기의 성능을 기존의 내장형 연산기들과 비교하면 다음과 같다. 표에 나타난 성능은 이상적인 경우를 가정한 것으로, 파이프 라인을 사용하여 여러 사이클에 걸쳐 연산하도록 설계하는 경우에는 컨트롤이 복잡해지고 익셉션이 발생하거나 데이터에 상호의존 관계가 있을 때에는 실제 성능이 떨어지는 일이 발생한다. 본 연산기는 한 사이클에 누적 연산까지 완료하도록 설계되었으므로 앞서 말한 성능 저하 현상이 발생하지 않는다.

표 3-1. 성능 비교

| | SIMD MAC Unit | TMS320C5X MAC Unit ^[2] | 32bit MAC ^[3] |
|---------------------------|--------------------|-----------------------------------|--------------------------|
| 동작 주파수 | 80MHz | 50MHz | 100MHz |
| 공정 | 0.35 μ m 표준셀 | 0.8 μ m full custom | 0.35 μ m full custom |
| Latency (16bit 기준) | 1 cycle | 3 cycle | - |
| 누적 저장기 내장 여부 | O | O | O |
| Datapath Width | 64bit | 32bit | 32bit |
| SIMD 지원 | 4 16bit 2 32bit | X | X |
| 초당 최대 연산 가능 횟수 (16bit 기준) | 320Mops | 50Mops | 100Mops |

SIMD 방식의 연산기는 현재 멀티미디어 데이터 처리에 대한 수요가 증가하면서 고성능 연산기의 한 방식으로 각광을 받고 있다. 그리고 점차 내장형 기기에서 대용량의 멀티미디어 데이터를 처리해야 하는 일이 늘어나면서 내장형 프로세서에 알맞은 간단한 구조의 소면적 SIMD 연산기에 대한 연구가 필수적이라고 하겠다.

본 논문에서는 일반적인 Booth 알고리즘을 이용한 곱셈기의 구조를 토대로 SIMD 연산이 가능한 MAC 연산기의 구조를 제안하였고, 30만개의 테스트 벡터를 인가하여 올바른 연산을 수행함을 확인하였다. 또한 HDL을 이용하여 하드웨어를 설계하여 최악조건에서 80MHz의 동작 주파수를 얻었다.

본 논문에서 제안한 연산기는 기존의 SIMD 연산기들에 비하여 핵심동작만을 간단한 구조로 수행할 수 있도록 구현되었고, 고성능의 SIMD 연산기에 비하여 면적이 작으면서도 일반적인 DSP 연산기들에 비하여 성능이 월등히 높으므로 멀티미디어 데이터 처리를 목적으로 하는 고성능의 내장형 프로세서나 DSP 프로세서에 내장하는 것이 적합할 것이다.

참고문헌

- [1] Israel Koren, *Computer Arithmetic Algorithms*, Prentice-Hall, pp. 73~77, 86~91, 99~123, 1993
- [2] *Buyer's Guide to DSP Processors*, Berkeley Design Technology Inc. pp. 351~361, 449~472, 1994
- [3] Bum-Sik Kim, Dae-Hyun Chung, and Lee-Sup Kim, "A New 4-2 Adder and Booth Selector for Low Power MAC unit", *Proceeding of International Symposium on Low Power Electronics and Design*, Monterey, pp. 100-103, 1997
- [4] Martin S. Schmoockler, Michael Putrino, Charles Roth, Mukesh Sharma, Anh Mather, Jon Tyler, Huy Van Nguyen, Mydung N. Pham, and Jeff Lent, "A Low-power, High-speed Implementation of a PowerPC™ Microprocessor Vector Extension", *Computer Arithmetic*, 1999. *Proceedings. 14th IEEE Symposium on*, 1999