

복소수 데이터 처리가 가능한 멀티미디어 프로세서용 고성능 연산회로의 하드웨어 설계

최병윤*

Hardware Design of High Performance Arithmetic Unit with Processing of Complex Data for Multimedia Processor

Byeong-yoon Choi*

Department of Computer Engineering, Donggeui University, Busan 47340, Korea

요 약

본 논문에서는 멀티미디어용 알고리즘을 고속으로 처리하기 위한 고성능 연산 회로를 설계하였다. 3단 파이프라인 구조로 동작하는 연산회로는 4개의 16-비트× 16-비트 곱셈기의 효율적인 구성, 캐리 보존 형식 데이터에 대한 새로운 부호 확장 기법과 다수 개의 부분 곱셈 결과의 통합과정에 부호 확장을 제거하는 교정 상수 기법을 사용하여 복소수 데이터와 가변 길이 고정 소수점 데이터에 대한 38개의 연산을 처리할 수 있다. 설계한 프로세서는 45nm CMOS 공정에서 최대 동작 속도는 300 MHz이며 약 37,000 게이트로 구성되며 300 MCOPS의 연산 성능을 갖는다. 연산 프로세서는 높은 연산 속도와 응용 분야에 특화된 다양한 연산 지원으로 멀티미디어 프로세서에 효율적으로 응용 가능하다.

ABSTRACT

In this paper, a high-performance arithmetic unit which can efficiently accelerate a number of algorithms for multimedia application was designed. The 3-stage pipelined arithmetic unit can execute 38 operations for complex and fixed-point data by using efficient configuration for four 16-bit by 16-bit multipliers, new sign extension method for carry-save data, and correction constant scheme to eliminate sign-extension in compression operation of multiple partial multiplication results. The arithmetic unit has about 300-MHz operating frequency and about 37,000 gates on 45nm CMOS technology and its estimated performance is 300 MCOPS(Million Complex Operations Per Second). Because the arithmetic unit has high processing rate and supports a number of operations dedicated to various applications, it can be efficiently applicable to multimedia processors.

키워드 : 연산회로, 복소수 데이터, 새로운 부호 확장 기법, 가변길이 데이터

Key word : arithmetic unit, complex data, new sign extension scheme, variable-length data

접수일자 : 2015. 11. 23 심사완료일자 : 2015. 12. 24 게재확정일자 : 2016. 01. 04

* **Corresponding Author** Byeong-Yoon Choi (E-mail:bychoi@deu.ac.kr, Tel:+82-51-890-1706)

Department of Computer Engineering, Donggeui University, Busan 47340, Korea

Open Access <http://dx.doi.org/10.6109/jkiice.2016.20.1.123>

print ISSN: 2234-4772 online ISSN: 2288-4165

©This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.
Copyright © The Korea Institute of Information and Communication Engineering.

I. 서 론

디지털 신호 처리, 컴퓨터 비전, 디지털 통신 등의 다양한 분야에 응용할 수 있는 멀티미디어 신호 처리는 복잡한 신호 처리 동작과 다양한 유형의 데이터에 대한 효율적인 처리를 필요로 한다[1,2]. 이러한 요구 조건을 만족하려면 범용 연산 기능과 함께 응용 분야에 특화된 복소수 데이터 처리와 서브워드(subword) 데이터의 벡터 처리가 바람직하다. 이러한 조건과 적은 면적을 만족하는 멀티미디어 프로세서는 연산의 유형에 따라 재구성 가능한 구조 또는 reduced instruction set computer(RISC) 혹은 very long instruction word (VLIW) 등의 범용 프로세서와 고정 소수점과 복소수 데이터를 효율적으로 처리할 수 있는 전용 연산 회로로 구성된 프로그램 가능한 프로세서 구조로 나누어진다[3].

멀티미디어 신호 처리의 핵심 연산인 가변 크기 데이터에 대한 곱셈을 지원하는 기존 연산회로는 큰 크기의 곱셈기(예: 32-비트 * 32-비트 곱셈기)를 구성한 후 데이터 크기와 연산의 종류에 따라 적절한 제어신호가 생성되어 원하는 크기의 곱셈 결과를 생성하도록 내부 분할하고, 중간 곱셈 결과를 다시 결합하는 구조이다[4-6]. 이러한 방식은 다양한 크기의 곱셈을 처리하려면 내부 분할, 배선 제어, 부분 곱 생성과 부호 확장 처리가 복잡하므로 확장이 어려운 단점이 존재한다.

본 논문에서 제시하는 연산회로는 4개의 16-비트 * 16-비트 부호/무부호 곱셈기를 사용하여 캐리 보존 형식(carry-save form)의 중간 결과를 생성한 후, 연산 유형과 데이터의 크기에 따라 4개의 곱셈 결과를 가중치에 따라 적절히 스케일링하여 결합하는 방식으로 원하는 크기의 곱셈 연산을 수행하도록 하여 기존 방식의 복잡한 배선과 확장 문제점을 해결하였다. 또한 캐리 보존 형식의 곱셈 중간 결과에 대한 새로운 부호 확장 비트 생성 방안과 중간 단계의 여러 개의 곱셈 결과를 통합하는 과정에 부호 확장을 제거하는 교정 상수 기법을 사용하여 빠른 동작과 면적 최적화가 가능하다.

본 논문의 구성은 2장에서는 가변 크기 데이터에 대한 MAC 연산의 병렬 처리 방안을 제시하고, 3 장에서는 멀티미디어용 연산회로의 하드웨어 설계를 기술하였으며, 4장에서는 설계 검증 및 성능 분석을 하였으며, 마지막으로 결론을 기술하였다.

II. 가변 크기 데이터에 대한 MAC 연산의 병렬 하드웨어 구현 방안

멀티미디어 신호 처리 분야의 핵심 연산은 다양한 크기의 데이터에 대한 곱셈과 multiply-and-accumulate (MAC) 연산이다. 그림 1은 다양한 크기의 데이터에 대한 3가지 유형의 병렬 곱셈을 나타낸다.

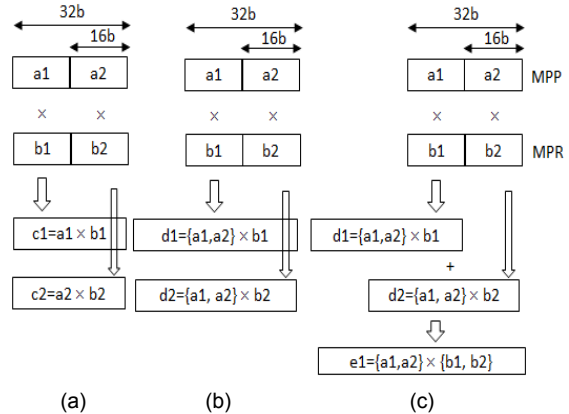


Fig. 1 Three types of parallel multiplications (a) two 16-bit * 16-bit multiplications (b) two 32-bit * 16-bit multiplications (c) a 32-bit * 32-bit multiplication

다양한 크기의 데이터에 대해 병렬 곱셈과 MAC 연산을 하드웨어를 공유하며 구현하는 방안은 크게 두 가지 방안이 있다.

첫 번째 방안은 분할 가능 곱셈기를 사용하는 방안이다[5,6]. 곱셈 연산의 병렬 처리 구조 특성을 활용하여 최대 크기의 데이터를 처리할 수 있는 하드웨어를 갖추고, 연산의 유형과 데이터 크기에 따라 내부를 적절히 분할해서 원하는 연산을 수행하는 방안이다. 그림 2는 16-b*16-b 곱셈기 하드웨어를 2개의 8-b*8-b 곱셈기 혹은 1개의 16-b*16-b 곱셈기로 활용하기 위한 개념적인 구조를 나타낸다. 그림 2에서 검정색 원은 부분곱의 0 또는 1인 하나의 비트를 나타내고, 흰색원은 0을, n으로 둘러싸인 흰색원은 부분곱의 2의 보수 구현 시 1의 보수에 더해지는 1을 나타내며, e로 둘러싸인 흰색원은 부분곱의 부호 확장을 나타낸다. 이 방안은 연산의 종류에 따라 입력 비트의 위치 조정, 부분곱 생성 회로의 변경 등이 필요하므로, 연산에 따라 복잡한 배선 제어가 필요한 단점이 있다.

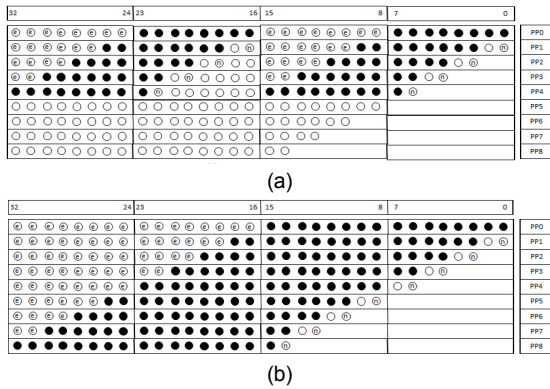


Fig. 2 Two configurations of 16-bit x 16-bit partitioned multiplier (a) two 8-bit multipliers (b) a 16-bit multiplier

두 번째 방안은 작은 크기의 다수 개의 곱셈을 병렬로 생성한 후, 중간 결과를 가중치(weight)에 따라 스케일링(scaling)하고 부호 확장하여 더해 원하는 결과를 얻는 병렬 곱셈-병합(parallel multiply and scaled merge) 방식이다. 그림 3은 16-비트 곱셈기를 병렬로 생성한 후 중간 결과를 적절한 비트 수 만큼 스케일링하여 2개의 16-비트 곱셈 혹은 1개의 32-비트 곱셈을 구현하는 동작을 나타낸다. 이 방식은 분할 곱셈 방식에 비해 제어가 단순하고 확장이 용이하다. 단, 이 방식을 직접적으로 구현하는 기존 방식[4]는 작은 크기의 곱셈에서 캐리 보존 형식이 아닌 완전한(non-redundant) 형태의 중간 곱셈 값을 구한 후 덧셈을 수행하므로 속도가 느려질 수 있는 문제가 있다. 본 연구에서는 속도 향상을 위해 캐리 보존 형식의 중간 곱셈 결과를 생성 후 더하는 방안을 사용하는데 이 경우 캐리 보존 형식 중간 결과에 대한 부호 확장 처리가 필요하다.

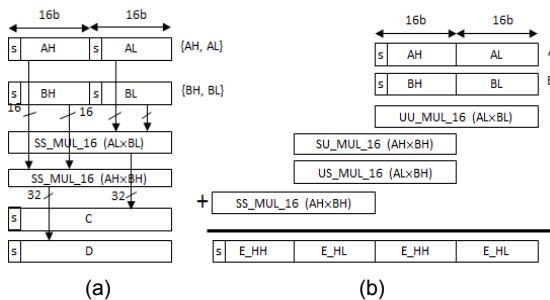


Fig. 3 Parallel multiplication and scaled merge scheme using 16-bit x 16-bit multipliers (a) two 16-bit multipliers (b) a 32-bit multiplier

III. 멀티미디어용 고속 연산 회로 설계

연산 회로의 설계 사양과 병렬 곱셈-병합 방식에 기반을 둔 연산 회로의 핵심 설계 기법을 기술한다.

3.1. 설계 사양

첫째, 멀티미디어 신호 처리 분야에서 핵심 연산인 MAC과 곱셈 연산을 다양한 크기의 데이터에 대해 고속으로 처리한다. 입력값으로 0 또는 1을 사용하여 덧셈 기능도 일부 지원한다. 둘째, 부호 및 무부호 16비트 x 16비트 곱셈기를 4개를 준비하고, 이를 조합하여 다양한 크기의 곱셈 결과와 다수 개의 작은 크기의 곱셈을 병렬로 계산하는 벡터 처리를 지원한다. 셋째, 멀티미디어 연산 시 오버플로와 언더플로 발생 시 중단이 없는 실시간 처리를 위해 포화 연산과 반올림 연산을 지원한다. 넷째, Fast Fourier Transform(FFT) 등의 응용을 위해 복소수 곱셈과 관련 연산도 지원한다. 다섯째, 3개의 64-비트의 입력을 받아서 3단 파이프라인 구조로 연산을 처리한다.

3.2. 지원하는 연산

그림 4는 데이터와 함께 제공되는 연산 코드 형식을 나타내며, 표 1은 5가지 형식으로 구분된 38개의 연산을 나타낸다.

Major_OP (3b)	Minor_OP (4b)	Control fields(7b) (sat, round, scale, acc, etc.)
---------------	---------------	--

Fig. 4 Format of operation code

Table. 1 Subset of operations

type	arithmetic type	number of operations
010	Complex_MAC	14
011	High_Prec_MAC	12
100	Single_Prec_MAC	4
101	Vector_Half_Word MAC	6
111	I_SAD	2

3.3. 16비트 x 16비트 부호/무부호 곱셈기를 사용한 MAC 구현 방안

본 연산 회로는 4개의 16비트 x 16비트 부호/무부호 곱셈기의 결과를 적절하게 병합하여 다양한 크기의 다수 개의 MAC을 구현한다. 그림 3의 32-b x 32-b 곱셈을

구현하기 위해 4개의 16-b × 16-b 부호 및 무부호 곱셈을 필요로 한다. 단, 32-비트 데이터가 부호화된 데이터라도 세부 16-비트 데이터는 상·하 위치에 따라 무부호 데이터가 될 수 있다는 점이다. 즉, UU_MUL_16은 승수와 피승수가 모두 무부호 데이터인 16-비트 곱셈을 나타내고, SU_MUL_16은 승수가 부호화 데이터이고, 피승수는 무부호 데이터인 16-비트 곱셈을 나타낸다. 따라서 본 연구의 세부 모듈인 16-비트 곱셈 회로는 기존의 곱셈기와 달리 승수와 피승수가 모두 무부호와 부호화 데이터가 될 수 있으므로 각 입력에 부호 제어기 필요한 범용 곱셈기가 필요하다[7]. 그림 5는 4개의 16-비트 범용 곱셈기를 사용한 32-b × 32-b 곱셈기의 단순화한 구조를 나타낸다. 각 곱셈기(MUL_16)는 빠른 연산 속도와 다음 단계의 중간 곱셈 데이터의 부호 확장 처리를 위해 33-비트 캐리 보존 형식의 결과와 부호 비트(MUL_sign)를 함께 출력한다.

3.4. 캐리 보존 형식 결과에 대한 부호 확장 기법

그림 5의 곱셈기에서 해결되지 않은 문제는 캐리-보존 형식의 곱셈 결과에 대한 부호 확장 처리이다. 일반적인 부호화 데이터의 경우 부호 비트의 단순한 반복 확장을 통해 처리가 가능한데[8-10], 캐리-보존 형식의 데이터의 경우 부호 비트를 결정하려면 하위 비트 순차적인 덧셈이 필요하다는 문제가 있다.

단, 곱셈의 경우 결과 출력 부호(MUL_sign)를 입력 데이터의 부호와 곱셈 연산 종류로 바로 결정할 수 있으므로, 캐리 보존 형식 출력의 부호 비트(최대 결과 범위보다 1-비트 상위 비트, 즉, 무부호 16-비트 곱셈의 기준 33-번째 비트), CV[n-1], SV[n-1]과 출력 부호(MUL

_sign) 값을 사용하여, 캐리 보존 형식 곱셈 결과에 대한 부호 확장 비트를 생성하는 방안을 제시한다. 그림 6은 곱셈 연산 종류와 입력의 부호 비트로 곱셈 결과 부호(MUL_sign)가 양수인 조건에서 캐리 보존 형식의 최상위 부호 비트 위치 값(CV[n-1], SV[n-1])이 {1, 1}인 경우를 나타낸다. 이 경우에 캐리 보존 데이터의 하위 비트 위치의 덧셈에서 발생하는 캐리, C_T는 MUL_sign = 0(양수) 조건을 만족하려면 0이 되어야 한다 (MUL_sign = CV[n-1] ⊕ SV[n-1] ⊕ C_T). 이러한 경우 올바른 부호 값(MUL_sign = 0)의 확장을 위해, 캐리 보존 형식의 상위부가 {1, 0} 혹은 {0, 1}의 값이 반복되는 형태로 상위 비트 방향으로 확장된다. 즉, 이러한 조건의 경우 비트 (n-1) 위치에서 CV[n-1]와 SV[n-1], C_T 덧셈에서 발생한 캐리는 계속 상위 비트에 캐리를 발생시켜, 캐리 보존 결과를 더한 곱셈 결과에 대해 양의 부호 0의 부호 확장 결과를 등가로 만들 수 있다. 유사한 방식을 출력 부호(MUL_sign)와 캐리 보존 데이터의 최상위 비트의 모든 조건에 적용하면, 부호 확장 비트는 식(1)과 같다. 여기서, n-1은 부호 비트 위치, n은 새로운 부호 확장 비트 위치를 나타낸다.

$$\begin{aligned}
 \text{MUL_SV}[n] &= \text{MUL_sign} + \\
 &\quad \text{MUL_sign}' \cdot (\text{MUL_CV}[n-1] + \text{MUL_SV}[n-1]) \\
 \text{MUL_CV}[n] &= \text{MUL_sign} \cdot \\
 &\quad (\text{MUL_CV}[n-1] \cdot \text{MUL_SV}[n-1]) \quad (1)
 \end{aligned}$$

식(1)에서 생성된 곱셈기의 캐리 보존 형식에 대한 부호 확장 비트 {MUL_CV[n], MUL_SV[n]}은 1-비트 쌍으로 유지되며 본 연구의 여러 가지 곱셈기의 통합 과정에 부호 확장 비트로 사용된다.

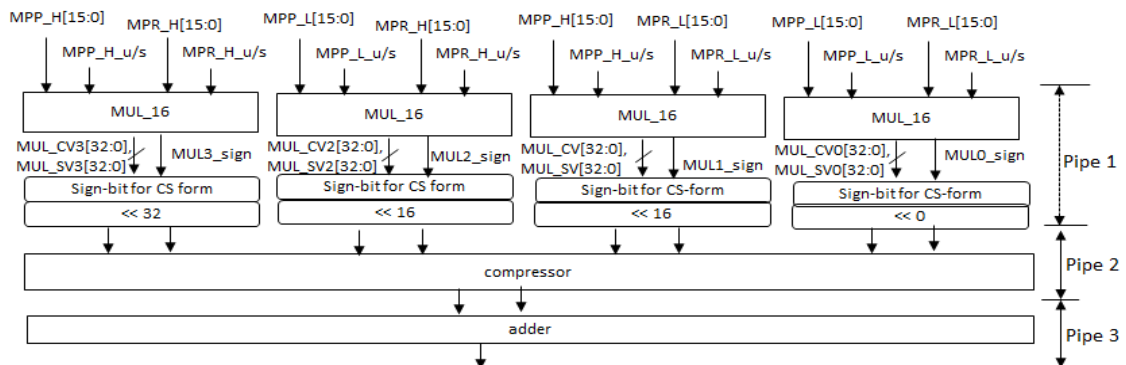


Fig. 5 Block diagram of 32-bit multiplier using four signed/unsigned 16-bit multiplier

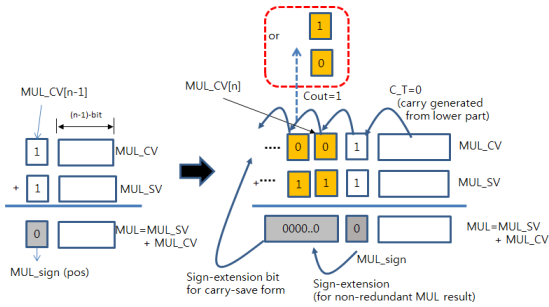


Fig. 6 Sign-extension bit generation for carry-save form

3.5. 복소수 데이터에 대한 연산

복소수 곱셈 연산은 그림 7과 같이 16-비트인 허수부와 실수부에 대해 각각 2개의 16-비트 곱셈과 덧셈 혹은 뺄셈의 조합으로 구현된다. 단, 누적 연산의 경우 64-비트 입력 레지스터의 상위 32-비트와 하위 32-비트와 더한 후, 포화(saturation) 연산 처리를 한 후 최종 결과를 얻게 된다. 또한 기존 연산을 약간 변형해서 공액(conjugate) 값도 계산한다.

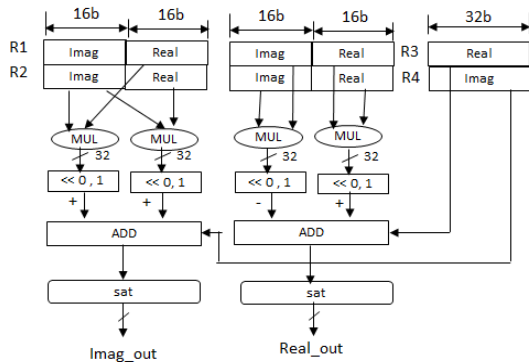


Fig. 7 Structure of complex multiply operation

3.6. compressor 단계에서 부호 확장 제거 기법

본 연산 회로는 38가지의 다양한 연산을 지원하므로 16-비트 곱셈 결과를 적절히 마스킹하고 가중치로 스케일링 처리를 하여, 다수개의 중간 곱셈 결과를 더하거나 빼게 되는데, 이때 캐리 보존 형식의 데이터에 대해 2의 보수 처리와 부호 확장을 통한 통합이 필요하다. 본 연구에서는 참고문헌[10]에서 제안된 방식을 변형해서 다수 개의 캐리 보존된 곱셈 결과를 부호 확장하여 더하는데 적용하였다. A4가 부호 비트인 경우 식(2)이 성립한다는 원리에 기반을 두고 있다. 즉, 부호 비트를 확

장하는 대신에 부호 비트 위치에서부터 1을 확장하고 기존 부호의 반전 비트((A4)')를 더한다.

$$A_4 A_4 A_4 A_4 = (1111 + (A_4)') \bmod 16 \quad (2)$$

그림 8은 32-b × 16-b 곱셈기에 적용한 기법을 나타낸다(①). S는 3.4절에서 제시한 캐리 보존 형식에 대한 새로운 부호 확장 비트로, 식(2)에 따라 기존 부호 확장 비트를 1의 연속된 비트 값과 부호 비트의 반전을 만든다(②). 부호 비트 위치의 2개의 1의 값을 더하면 캐리가 발생하는데, 이것은 상위의 1의 비트 열을 1개로 단순화시킨다(③).

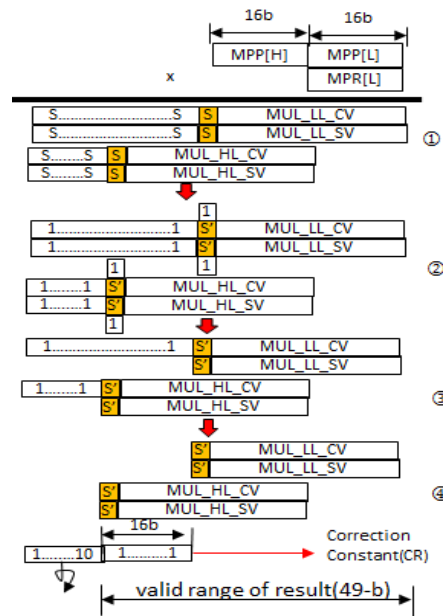


Fig. 8 Sign extension and correction constant generation scheme in 32-b × 16-b multiply operation

최대 결과 범위(무부호 곱셈의 경우 결과 부호 포함: 49-비트)를 초과하는 비트 위치 값을 저장할 필요가 없다. 마지막으로 2개의 1의 비트열을 더할 경우 상위부의 1은 제거되고 16-비트의 연속된 1이 남게 되는데, 이 값을 교정 상수(correction constant)로 명명하였다(④). 이 교정상수 값은 곱셈 연산과 중간 결과의 스케일링 조건에 따라 다른 값이 생성되는데, 곱셈 결과의 누적 덧셈 과정에 보정용 덧셈 항으로 더해져 부호 확장을 제거할 수 있다.

3.7. 고성능 연산 회로의 하드웨어 설계

설계한 연산 회로는 38가지 연산을 지원하고 높은 동작 속도를 위해 3단 파이프라인 구조로 구성된다. 그림 9는 전체 연산 회로의 구조를 나타낸다. 단 두 개의 64-비트 데이터를 8-비트 단위로 뺄셈을 한 후 그 값의 절대치를 구하는 SAD 연산은 별도의 SAD_unit 하드웨어에서 구현하고, 누적합 연산은 기존 연산회로 단계 2의 compressor 하드웨어를 활용한다. 연산을 정의하는 OPcode는 제어 회로의 파이프라인 단계를 지나가면서 적절한 제어신호를 생성하여 연산 회로 데이터패스를 제어한다. ID 단계는 제어용 RISC 프로세서의 명령어 디코드 단계에 통합된다.

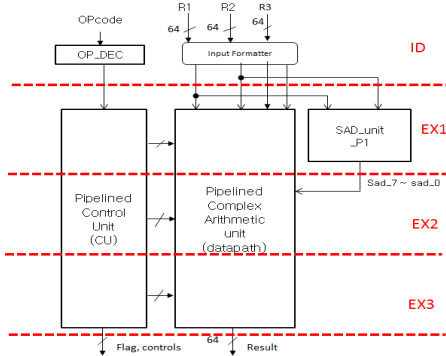


Fig. 9 Block diagram of pipelined arithmetic unit

그림 10은 연산 회로의 데이터패스를 나타낸다. 연산 회로의 핵심 연산은 누적 기능을 갖는 1개의 32b × 32b MAC, 2개의 32-b × 16-b MAC/16-b 복소수 MAC, 2개의 16-b × 16-b MAC 동작 등이다. 파이프라인 단계 2에서 캐리 보존 곱셈 결과에 대한 누적 덧셈 결과 과정에 반올림을 반영한 최종 캐리 보존 형식의 출력을 생성한다. 단, 하드웨어 공유를 위해 32-b × 32-b 곱셈의 경우 compressor 단계에서 먼저 2개의 32-b × 16-b 곱셈 결과에 대한 누적 합을 중간 결과로 구한 후, 중간 값을 스케일링한 후 더하는 방식으로 구현된다. 파이프라인 단계 3에서 포화 연산 기능을 갖는 덧셈을 하여 64-비트 결과 혹은 2개의 32-비트 결과를 생성한다. 32b × 16b MAC의 경우 하위 16-비트에서 반올림을 하고, 16-비트 우측 스케일링으로 32-비트 결과를 생성한다.

IV. 설계 검증 및 성능 분석

설계한 연산 회로에 대한 검증을 단순화하기 위해 두 가지 방식으로 Verilog-2001 HDL로 모델링하였다. 논리 합성을 위해 기능 수준으로 모델링함과 동시에 38개의 연산에 대해 곱셈 연산자(*), 덧셈 연산자(+)와 시프트 연산자(<<,<<<) 등을 조합하여 동작 수준의 시뮬레이션 모델을 별도로 작성하여, 검증 과정에 두 개의

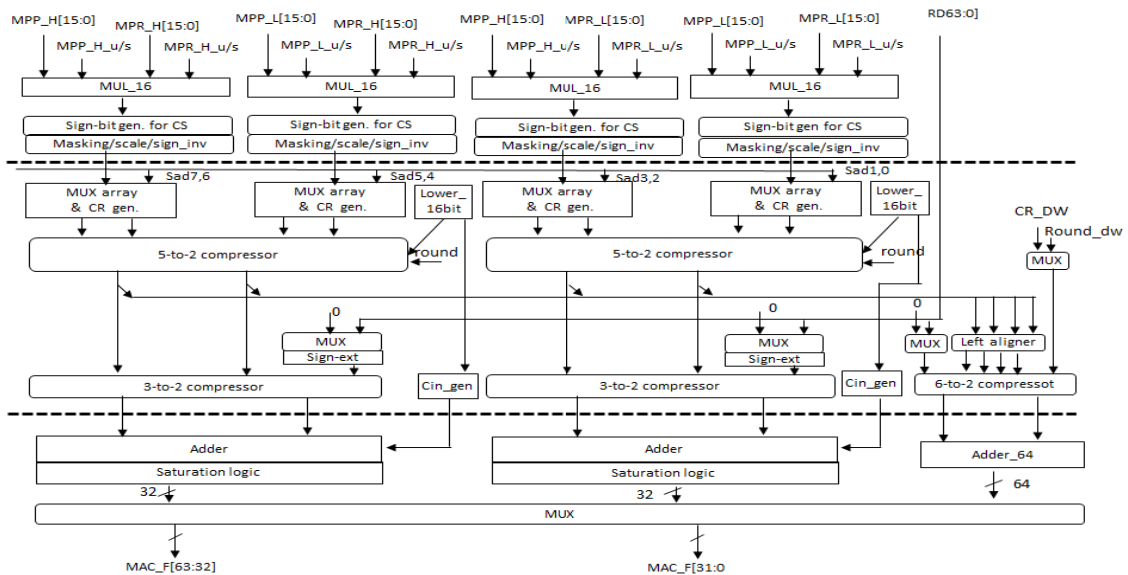


Fig. 10 Block diagram for datapath of high performance arithmetic unit

결과가 일치하는 지 쉽게 판단할 수 있도록 하였다.

그림 11은 1-비트 스케일링(<<1)을 포함한 복소수 곱셈에 대한 Modelsim 검증 결과를 나타낸다. 설계한 연산 회로는 45nm CMOS 표준셀 공정에서 Synopsys 합성틀로 합성한 결과 약 300 MHz 동작 주파수와 약 37,000 게이트를 가지고 있음을 확인하였다. 그리고 연산 회로는 캐리 보존 데이터에 대한 효율적인 부호 확장 비트 생성과 여러 개의 캐리 보존 곱셈 결과에 대한 덧셈 과정에 교정 상수를 통한 효율적인 부호 확장 제거 방안을 사용하여 면적을 최소화하고 동작 속도를 개선하였다. 또한 오버플로와 언더플로에 의한 예외 처리 문제를 배제하기 위해 포화 연산을 지원하며, 정밀도 향상을 위해 반올림 연산 기능을 지원한다. 현재 설계된 연산 회로는 벤처 회사에 제공되어 반도체 Intellectual Property(IP)로 사용되고 있다. 표 2는 설계한 연산회로에 대한 특성을 나타낸다. 설계한 연산 회로는 38개의 연산을 지원하며, 45nm 공정 조건에서 초당 약 3억 개의 복소수 연산(300 MCOPS) 혹은 병렬 처리로 초당 6억 개의 16-b × 16-b 와 32-b × 16-b 곱셈 연산이 가능하므로 고성능을 요구하는 멀티미디어 분야에 효율적으로 응용 가능하다. 설계한 회로는 n-비트 곱셈만을 기준으로 할 경우 먼저 작은 크기(r-비트)의 수정된 Booth 구조 배열(array) 곱셈기에서 캐리 보존 형식의 중간 결과를 생성하고, (n/r)²개의 부분곱을 (3,2) CSA 트리로 2개의 중간 결과를 생성한 후, 고속 carry lookahead adder(CLA)로 2n-비트 결과를 더하므로, 계산 복잡도는 $O(r/2 + \log((n/r)^2)/\log(3/2) + \log 2n)$ 로 표현된다.

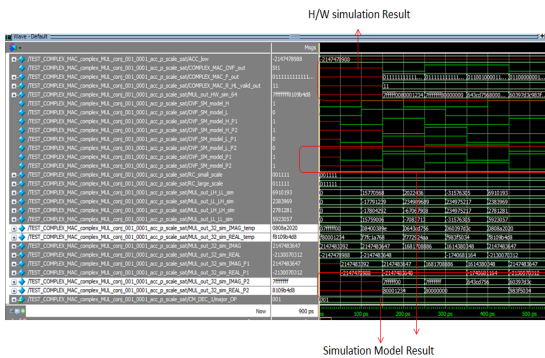


Fig. 11 Modelsim simulation waveform for complex multiply and accumulation(CMAC) operation

표 3은 본 연구에서 설계한 복소수 연산 프로세서와 유사한 기능을 갖는 기존 프로세서 혹은 곱셈기와 특성을 비교한 결과이다. 참고 문헌[11]의 경우 중간 결과가 캐리 보존 형식의 결과가 아닌 완전한(non-redundant) 곱셈 결과를 사용하므로 속도가 느리며 무부호 곱셈에만 적용할 수 있다는 제약 사항이 있고, 참고 문헌 [6]은 확장성에 문제가 있다. 본 논문의 연산회로는 우수한 확장성과 효율적인 부호 확장, 다양한 연산 지원으로 멀티미디어에 특화된 알고리즘을 효율적으로 구현할 수 있다.

Table. 2 Characteristics of arithmetic unit

number of pipeline stage	3
number of operations	38
variable-size multiplication scheme	PMSM_CS (parallel multiply and scaled merge scheme with carry-save midterm MUL results)
sign extension scheme for multi-operand(MUL) data	sign-extension elimination using correction constant(CR)
sign extension scheme for carry-save form multiplier data	sign extension bit generation using MUL result sign and sign(MSB) bits of carry-save result
time complexity for n-bit multiplier (with r-bit smaller MUL)	$O(r/2 + \log((n/r)^2)/\log(3/2) + \log 2n)$
number of gates	about 37,000
operating frequency	300 MHz@45nm CMOS process
performance	300 MCOPS(Million Complex Operations per Second)

Table. 3 Performance comparisons

processors	# of gates (# of tr./ 4)	architectural characteristics	# of operations
Partition_blk_MUL [11]	7,536 (30,144/4)	PMSM_NR* using eight 8-b×8-b multiplier	1 (64-b MUL)
Partition_P MUL[6]	-	PAMUL* using a 64-b×64-b multiplier	3 (64-b,32-b, 16-b MUL)
Proposed	37,000	PMSM_CS* scheme using four 16-b×16-b multiplier	38

*PMSM_CS(parallel multiply and scaled merge with carry save midterm MUL result) scheme

*PMSM_NR(parallel multiply and scaled merge with non-redundant midterm MUL result) scheme

**PAMUL(partitioned multiplier) scheme

V. 결론

설계한 연산 회로는 다양한 크기의 정수 및 복소수 MAC 연산, 8개의 8-비트 데이터의 뺄셈의 절대치에 대한 누적합을 구하는 SAD 명령 등 멀티미디어 분야에 특화된 38개의 연산을 지원한다. 그리고 성능 향상을 위해 가변 크기 데이터 MAC 동작을 3가지 관점에서 기존 방식을 개선하였다. 첫째 가변 데이터 크기를 지원하기 위해 4개의 범용 16-b × 16-b 곱셈기를 사용한 캐리 보존 형식의 중간 결과를 갖는 병렬 곱셈-병합 방식을 사용하였다. 둘째로 캐리 보존 형식의 곱셈 결과에 대해 효율적인 1-비트의 쌍으로 구성된 부호 확장 비트 생성 방안을 제시하였다. 셋째, 다수 개의 캐리 보존 형식의 곱셈 결과를 스케일링한 후 병합하는 연산 과정에 교정 상수를 통한 부호 확장 제거 방안을 제안하였다. 그리고 디지털 신호 처리에서 요구되는 포화 연산과 반올림 기능을 지원하고 있다. 설계한 회로는 45nm CMOS 표준 셀 공정 조건에서 약 37,000 개의 게이트로 구성되며, 3단 파이프라인 방식으로 2개의 32-b × 16-b MAC/ 16-b × 16-b MAC 혹은 1개의 32-b × 32-b MAC 혹은 1개의 복소수 곱셈 연산을 지원할 수 있어서, 최대 300 MCOPS의 연산 성능을 가지고 있다. 설계된 고성능 연산 회로는 멀티미디어 분야에 특화된 다양한 연산 명령을 내장하고 있어서 반도체 IP로 멀티미디어 프로세서에 활용이 가능하다고 판단된다.

ACKNOWLEDGMENTS

This work was supported by Dong-eui University Grant(2015AA070). The IDEC CAD tools were partially used to verify operation of circuit and algorithm.



최병윤(Byeong-Yoon Choi)

1985년 2월 : 연세대학교 전자공학과 졸업
 1987년 2월 : 연세대학교 공학석사
 1992년 8월 : 연세대학교 공학 박사
 1993년 3월 ~ 현재 : 동의대학교 컴퓨터공학과 교수
 *관심분야 : 마이크로프로세서 설계, SoC 설계, 임베디드 시스템, IP 설계

REFERENCES

- [1] Ruby B. Lee, "Subword Parallelism with MAX-2," *IEEE Micro*, vol.16, no. 4, pp.51-59, August 1996.
- [2] Ruby B. Lee, "Accelerating Multimedia with Enhanced Microprocessors," *IEEE Micro*, vol.15, no.2, pp.22-32, April, 1995.
- [3] QualComm, *Hexagon V2 Programmer's Reference Manual*, 80-NB419-1 Rev.A, August 2011.
- [4] Israel Koren, *Computer Arithmetic Algorithms*, ch.5-6, CRC Press, 1993.
- [5] Aamir Alam Farooqui, "VLSI Arithmetic for Media Signal Processing," Ph.D dissertation, ECE department, UC Davis, 2000.
- [6] Hyuk-Jun Lee and Michael Flynn, "Designing a Partitionable Multiplier," Stanford University, Technical Report CSL-TR-98-772, October 1998.
- [7] Hesham Al-Twajjry and Michael Flynn, "Performance/Area Tradeoffs in Booth Multipliers," Stanford University, Technical Report CSL-TR-95-684, November 1995.
- [8] Alexander F. Tenca, Song Park, and Lo'al A. Tawalbeh, "Carry-Save Representation Is Shift-Unsafe: The Problem and Its Solution," *IEEE Transactions on Computers*, vol. 55, no.5, pp.630-635, May 2006.
- [9] Stuart F. Oberman, and Ming Y. Siu, "A High-Performance Area-Efficient Multifunction Interpolator," Proc. of the 17th IEEE Symposium on Computer Arithmetic(ARITH'05), pp.271-279, 2005.
- [10] M. Roorda, "Method to reduce the sign bit extension in a multiplier that uses the modified booth algorithm," *Electronics Letters*, vol.22. no.20, pp.1061-1062, 25th September 1986.
- [11] Christopher Fritz and Adly T. Farm, "The Interlaced Partition Multiplier," *IEEE Trans. on Computer*[online], no. 1, pp. 1, PrePrints, doi:10.1109/TC.2015.2481379, Available: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7274668>.