

고성능 DSP 아키텍처 설계에 대한 연구

윤성철, 허경희, 배성일, 강성호

연세대학교 전기전자공학과

전화: (02)2123-2775 / 팩스: (02)313-8053

The Research of High-Performance DSP Architecture

Sungchul Yoon, Kyunghoi Huh, Sungil Bae, Sungho Kang

Department of Electrical and Electronic Engineering, Yonsei University

E-mail : shkang@yonsei.ac.kr

Abstract

DSP is used for processing the digital data in such as the multimedia applications. Because the digital data of high rate is demanded more and more, high performance is increasingly required in DSP. In this paper, we discuss important issues for development of high performance DSP, analyze architectures of several commercial DSP chips, and propose a new architecture. Finally, we show that the new architecture has the highest performance.

I. 서론

최근의 놀라운 만한 반도체 공정기술과 높은 클럭 주파수의 영향으로 손톱크기만한 칩안에 멀티미디어 정보를 처리할 수 있는 회로를 집적할 수 있게 되었다. 그에 따라 사람들의 요구는 더욱 커지게 되고, 보다 강력한 연산능력에 의한 데이터의 빠른 처리가 더욱더 요구되고 있다. 그리고 마이크로프로세서의 성능이 매우 좋아져 빠른 연산에 의한 데이터처리가 가능하게 되었지만 DSP 알고리즘의 처리를 마이크로 프로세서가 한다면,

그 처리로 인한 시간 손실이 만만찮게 작용하는 면이 없지 않다. 그러므로 DSP의 성능향상에 대한 요구가 점점 커지고 있는 현실이다. DSP라는 것은 여러 가지 디지털 신호를 여러 알고리즘에 맞게 처리해주는 프로세서이다. 여기서 여러 알고리즘에는 시간공간의 데이터를 주파수공간으로 변환해 주는 FFT, 그리고 필요한 주파수 성분을 필터링해주는 여러종류의 필터, 또 에러정정코드를 디코딩해주는 Viterbi나 Turbo 디코딩등이 있다. 그리고 압축기술인 MP3나 G.726같은 알고리즘도 포함된다.

지금까지 DSP의 주된 아키텍처는 MAC (Multiply and Accumulator)가 하나인 Single MAC DSP였지만 앞에서의 성능향상에 대한 요구는 Single MAC의 한계를 극복해야 하는 현실을 만들고 있다. 물론 MAC이 두 개인 Dual MAC구조를 갖는 DSP 아키텍처도 이미 나와 있지만 성능 향상이 필요한 면이 없지 않다. 그래서 이 논문에서는 Dual MAC을 갖는 세 개의 아키텍처를 소개하고, 각 응용에 따른 성능비교를 한 후 그 결과를 토대로 아키텍처상의 장단점을 분석한다. 그리고 성능이 우수한 새로운 아키텍처를 제안한다.

II. DSP 처리함수의 종류

DSP(Digital Signal Processor)가 처리하는 함

수의 종류는 매우 다양하다. 단순한 곱셈, 덧셈에서부터 FFT, Digital Filter, 그리고 Viterbi decoding까지 상황에 따라서는 수십가지에 이른다. 마이크로프로세서나 마이크로 컨트롤러가 있는 시스템의 경우에는 DSP가 coprocessor로 동작을 하며, 복잡한 곱셈이나 많은 bit를 갖는 데이터의 덧셈같은 것을 수행함으로써 CPU의 수행 동작을 도와주는 역할을 하고, 통신에 사용되는 DSP의 경우와 같이 DSP가 시스템을 제어하면서, 통신에 필요한 함수를 직접 처리하는 역할을 한다. 그 중에서 이 논문에서 비교대상이 되는 DSP에 적용한 알고리즘들은 표1과 같다.[1]

표1. 적용알고리즘 목록

1. real 콘볼루션
2. complex 콘볼루션
3. 계수가 5개인 IIR 필터(Form I)-단정밀도
4. 계수가 5개인 IIR 필터(Form I)-배정밀도
5. 계수가 4개인 IIR 필터(Form II)-단정밀도
6. 계수가 4개인 IIR 필터(Form II)-배정밀도
7. 계수가 5개인 IIR 필터(Form II)-단정밀도
8. 계수가 5개인 IIR 필터(Form II)-배정밀도 (3-8은 모두 Biquad 필터)
9. Lattice Forward IIR 필터
10. Lattice Inverse IIR 필터
11. Radix 2 FFT butterflies
12. LMS Adaptive 필터-단정밀도
13. LMS Adaptive 필터-배정밀도
14. Viterbi 디코딩(8state)

이 밖에도 여러 알고리즘이 있지만 이번 연구에서는 표에 나타난 알고리즘이 대표적이고 거의 모든 연산종류와 유사성을 가질 수 있으므로 표의 알고리즘만을 고려하였다. 표에 나타난 알고리즘으로도 상당히 신뢰성을 가질 수 있는 아키텍처 구별이 가능하다.

MAC이 많을수록 성능이 향상될 수 있지만 MAC이 하나에서 두 개만 되어도 제어나 데이터 패스에 상당한 부담으로 작용이 되므로 세 개 이상을 쓰는 것은 오버헤드나 제어문제로 오히려 더욱 가격대 성능비가 떨어진다. 그렇지만 MAC을 두 개를 쓰는 것은 지금의 기술문제로 보아 위의 문제를 해결하면서 성능을 높이는 방법이 될 수 있다. MAC을 두 개 사용함으로써 콘볼루션

선이나 상관도를 구하는데 거의 두배의 성능을 볼 수 있다. 필터는 종류에 따라, 아키텍처의 영향을 상당히 많이 받는다. 예를 들면 LMS 필터의 경우 세 개의 데이터를 곱하는 연산이 있는데, 곱한 후 accumulator에 저장하는 동시에 곱셈 레지스터에 가져다 놓는 명령어를 지원하는 아키텍처가 좋은 것은 자명한 일이다.

III. 새로운 DSP 아키텍처의 구조와 성능

그림1은 새로운 DSP 아키텍처의 구조를 나타낸 블록다이어그램이다. 데이터와 프로그램 등의 주소를 생성하는 RPU, 직접적인 연산을 수행하는 CU, 그리고 코어내의 제어를 담당하는 SCU로 이루어져 있고, 다른 프로세서와의 연결을 담당하는 IU가 있다. SCU에는 명령어 캐쉬와 버퍼를 두어서 효과적인 제어기능을 수행하도록 하였다.

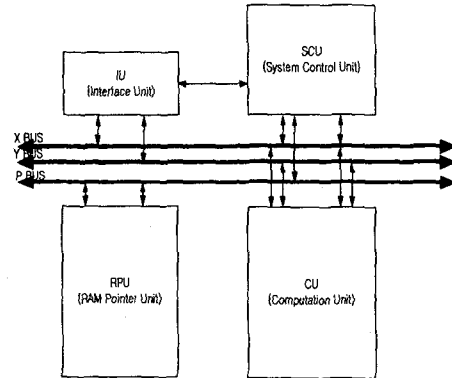


그림 1. 새 아키텍처의 블록다이어그램

그림2는 DSP 성능에 가장 큰 영향을 미치는 계산 유닛인 CU의 구조를 그린 것이다. 쉬프트를 하는 유닛을 ALU에 포함하고, 병렬로 또 하나의 쉬프트를 위치시킴으로써 Viterbi 연산을 하면서 산술 연산 동작을 병렬적으로 할 수 있도록 하였다. 표2는 새로운 아키텍처의 병렬가능 명령어 집합이다. RISC 적인 구조를 갖기 때문에 명령어 집합이 비교적 간단하다. split mode에서는 3 입력 ALU 연산과 2 입력 ALU 연산을 동시에 할 수 있다.

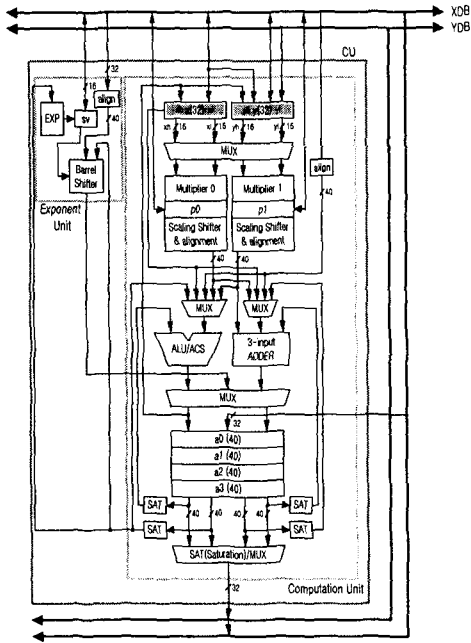


그림 2. CU(Computational Unit)의 구조

표 2. 새로운 아키텍처의 병렬 가능 명령어 집합의 대표적인 예

종류	Instruction set
MAC	add mpy(2) move
MSC	sub mpy(2) move
Viterbi	acs shifr/shifl move
Split	add3 add move

IV. Dual MAC DSP의 성능평가

네 개의 아키텍처 모두 Dual MAC을 갖는 아키텍처이고, 버스가 데이터버스, 프로그램버스로 나누어져 있다. Teak[4]과 DSP16000[2]은 load/store 구조를 갖는 반면, TMS320C55x[3]는 CISC적인 구조를 갖는다.[5]

TeakDSP는 32bit 버스가 4개 존재하는 것이 특징이다. dual MAC 구조에서 버스가 많은 것은 장점이지만 명령어 캐쉬나 명령어 버퍼가 없는 구조를 갖기 때문에 한계가 있다.

표3에서 보듯이 세 DSP core는 비슷한 성능을 보이지만 차이가 많이 나는 알고리즘도 존재한다.

같은 Dual MAC 구조를 가져도 차이가 나는 것은 당연히 아키텍처의 차이 때문이다. LMS adaptive 필터의 경우 곱셈결과를 다시 곱셈기의 입력으로 넣는 동작이 필요하다. 이 경우 DSP16000의 아키텍처가 매우 유리하다.

표 3. 네 DSP 아키텍처의 프로그램 실행 능력비교 (N:사이클타임)

DSP 표1의 알고 리즘	Teak	DSP 16000	TMS320 C55x	새로운 아키텍처
1	N/2	N/2	N/2	N/2
2	2N	2N	2N	2N
3	9N	7N	7N	7N
4	20N	17N	22N	17N
5	3N	3N	4N	3N
6	14N	12N	16N	12N
7	8N	5N	5N	5N
8	17N	16N	20N	16N
9	4N	4N	4N	4N
10	5N	5N	5N	5N
11	7N/2	7N/2	5N/2	5N/2
12	3N	2N	2N	2N
13	5N	4N	6N	4N
14	22N	26N	18N	18N

DSP16000에서는 곱셈결과를 다시 곱셈기의 입력으로 넣을지의 여부를 결정하는 상태레지스터의 상태 bit이 존재하기 때문이다.

TMS320C55x의 경우 CISC적인 아키텍처를 가지기 때문에 메모리에서 가져와 메모리로 다시 쓰는 연산에서는 매우 좋지만 버스가 16bit이므로 배정밀도에서는 단점을 보인다. 결과를 쓰는데 2 cycle이 걸리기 때문이다. 배정밀도 연산에서도 DSP16000이 장점을 보이고 있는데 이것은 버스가 32bit이고 한번에 연산결과를 쓸 수 있기 때문이다. 단정밀도에서도 상태레지스터의 상태 bit에 따라 두 개의 연산결과를 한꺼번에 메모리에 쓸 수 있는 DSP16000이 장점을 갖는다. 짝수와 홀수의 연속된 accumulator의 상위부분을 묶어서 벡터처럼 한번에 메모리에 갖다 쓸 수 있는 모드가 있기 때문이다. Teak의 장점은 메모리 주소 생성을 담당하는 유닛에 있다. 다양한 주소생성모드

표 4. 아키텍처 비교

내용 \ DSP	Teak DSP	DSP16000	TMS320C55x	새로운 아키텍처
버스	-3개의 32bit data bus -1개의 32bit program bus	-1개의 32bit data bus -1개의 32bit common bus	-3개의 16bit read bus -2개의 16bit write bus -1개의 32bit program bus	-3개의 32bit data bus -1개의 32bit program bus
Instruction Length	16bit and 32bit	16bit and 32bit	8 to 48bits	16bit and 32bit
Instruction Cache	None	31-entry(124bytes) cache	24 Kbytes cache	31-entry(124bytes) cache
Instruction Buffer	None	None	64byte buffer	64byte buffer
Dual MAC	-2 multipliers -3input ALU	-2 multipliers -2input ALU -3input ADDER	-2 MACs -2input ALU	-2 multipliers -3 input ALU -2 input ADDER
Multiplier input registers	4개의 16bit registers (x0,x1,y0,y1)	2개의 32bit registers (x,y)	Not specified	2개의 32bit registers (x,y)
BMU	ALU	Separated BMU	Shifter	Separated BMU
Accumulators	4	8	4	4
ALU inputs	abX, p0, p1 XDB	aX, y, p0, p1	CB,DB ACx shifter output	aX(공유), y, p0, p1
Data Transfer	M to x, y, abX, ALU input	M to x, y, p0, p1, aX	M to M, ACx, MAC input, ALU input, shifter input	M to x, y, p0, p1, aX

주소생성유닛에서 지원을 하지만 그리 복잡하지 않게 구현한 것이 장점이라 하겠다. 새로운 아키텍처는 버스를 데이터 버스를 세 개를 두어서 양 방향으로 동작하게 하여, 병렬 연산과 MAC 동작을 효과적으로 지원할 수 있도록 하였다. 명령어 캐쉬와 명령어 버퍼를 모두 두어서 최대한의 병목현상을 해결하고, 효과적으로 프로시저 제어를 할 수 있도록 하였다. 그리고 BMU를 따로 두어서 산술 연산과 비트별 연산을 동시에 할 수 있도록 하였다. 그리고 aX의 입력과 곱셈기의 입력을 공유함으로써 곱셈 후 다시 곱하는 경우, aX에 저장을 하면서 다시 곱셈기의 입력으로 들어갈 수 있는 모드를 지원하였다.

아키텍처가 제시한 알고리즘들에 대해서 가장 고성능을 발휘하는 것을 알 수 있다. 이제는 이 새로운 아키텍처를 실제로 구현하는 일을 진행할 것이다.

참고문헌

- [1] Richard J. Higgins, *Digital Signal Processing in VLSI*, Prentice Hall, 1990
- [2] DSP16000 Reference Guide
- [3] TMS320C55x DSP CPU Reference Guide
- [4] Teak™ DSP Core Architecture Spec.
- [5] Patterson & Hennessy, *Computer Organization & Design*, Morgan Kaufmann, 1998

V. 결론 및 향후과제

지금까지 새로운 고성능 DSP 아키텍처에 대해 논하였다. 네 개의 아키텍처 중에 새롭게 제안한