

論文98-35C-12-4

휴대 멀티미디어 응용을 위한 DSP 칩 설계 및 구현

(Design and Implementation of a DSP Chip for Portable Multimedia Applications)

尹 成 賢 * , 鮮 于 明 勳 *

(Sung Hyun Yoon and Myung Hoon Sunwoo)

요 약

본 논문은 휴대 멀티미디어 응용을 위한 고정 소수점 DSP(Multimedia Fixed-point DSP : MDSP) 칩 설계 및 구현에 관해 기술한다. MDSP는 멀티미디어 처리에 효율적인 명령어 집합을 가지며 SIMD, 벡터 프로세싱의 병렬처리 기술과 DSP 기술의 장점을 접목하여 설계되었다. MDSP는 한개의 데이터 경로가 목적에 따라 여러개로 분할될 때 8, 16, 32, 40 비트 등의 다양한 데이터 형태의 처리가 가능하며, 멀티미디어 응용영역에서 핵심적인 역할을 하는 MAC 연산을 한 사이클에 2개를 수행하여 성능을 향상시킨다. 새롭게 제안된 스위칭 네트워크와 Packing 네트워크는 MPEG 디코딩, 인코딩, 콘볼루션 등의 알고리즘 처리시 연산과 데이터 변환을 중첩시켜 성능을 향상시킨다. Verilog HDL 모델을 구현하였고 0.6 μm SOG 라이브러리(KG75000)를 이용하여 논리합성 및 시뮬레이션 하였다. 전체 게이트 수는 68,831개이며 MDSP는 30 MHz에 동작한다.

Abstract

This paper presents the design and implementation of a new multimedia fixed-point DSP (MDSP) core for portable multimedia applications. The MDSP instruction set is designed through the analysis of multimedia algorithms and DSP instruction sets. The MDSP architecture employs parallel processing techniques, such as SIMD and vector processing as well as DSP techniques. The instruction set can handle various data formats and MDSP can perform two MAC operations in parallel. The switching network and packing network can increase the performance by overlapping data rearrangement cycles with computation cycles. We have designed Verilog HDL models and the 0.6 μm Samsung KG75000 SOG library is used. The total gate count is 68,831 and the clock frequency is 30 MHz.

I. 서 론

1980년초 음성, 음향, 통신 및 영상신호의 디지털 처리를 위한 DSP(Digital Signal Processor) 칩의 개발로 다양한 분야에서 과거의 일반적인 마이크로 프

로세서로 구현하기 어려웠던 실시간 응용 시스템을 실현할 수 있었다. 특히 DSP 칩의 die 면적을 작게 하여 제조 원가를 절감하고 전력소모를 줄이기 위해 고정 소수점 DSP 칩을 여러 분야에서 많이 사용하고 있다. 최근 다양한 멀티미디어 정보처리, 고품질 통신 등을 위해 기존의 아날로그 시스템에서 사용되는 신호 처리 기능 대부분이 디지털화 되는 추세이며 따라서 프로그래밍이 가능한 DSP 칩의 중요성이 증가되고 있다.

* 正會員, 亞州大學校 電機電子工學府

(School of Electrical and Electronic Eng., Ajou Univ.)

接受日字:1998年7月20日, 수정완료일:1998年11月4日

기존의 DSP 칩^[1-4]은 주로 신호처리 전용으로 개발되었기 때문에 처리 데이터 패스가 16, 24 비트 등으로 고정되고 영상처리에 부적합하며 방대한 양의 데이터에 대해 많은 연산량을 요구하는 영상 및 멀티미디어 데이터를 실시간 처리할 수 없는 단점을 갖는다. 따라서 멀티미디어 데이터를 실시간 처리하기 위해 새로운 고성능 멀티미디어 DSP 칩들이 개발되고 있다. 이와같은 멀티미디어 DSP 칩들은 여러 가지 매체를 효율적으로 처리할 수 있도록 설계되어 좋은 성능을 낼 수 있으나 새롭게 설계해야 하기 때문에 개발 환경을 구축하는데 많은 시간과 노력이 필요하다.

멀티미디어 DSP 칩은 최근 몇몇 외국 회사에서 상용화하였거나 현재 시제품으로 제작중이다. 현재까지 개발되었거나 개발중인 멀티미디어 DSP 칩들은 MicroUnity사의 MediaProcessor^[56], Chromatic사의 Mpack 및 Mpack2^[78], Philips사의 TriMedia^[9], TI(Texas Instrument)사의 TMS320C62xx^[10,11], Analog Device사의 ADSP-2106x^[12] 등이 있다. 이외에도 고성능 범용 마이크로 프로세서에 멀티미디어 명령어를 추가한 경우에는 SUN사의 UltraSPARC^[13], Intel사의 ix86^[14], HP사의 PA-RISC^[15] 아키텍처가 있으며, 이들은 각각 VIS(Visual Instruction Set)^[16], MMX(Multi-Media Extensions)^[17], MAX(Multimedia Acceleration eXtensions) 및 MAX-2^[18] 명령어 집합을 추가하였다. 이들 이외에도 멀티미디어 확장을 위해 SGI사는 MIPS 프로세서에 MDMX(Mips Digital Media Extensions)^[19], DEC사는 MPEG2 인코딩을 위해 Alpha 프로세서에 MVI(Motion Video Instructions)^[20] 명령어 집합을 추가하였다.

위에서 언급한 멀티미디어 처리를 위한 칩들은 공통적으로 SIMD(Single Instruction Stream - Multiple Data Streams), 슈퍼스칼라, VLIW(Very Long Instruction Word), 멀티스레드, 벡터 프로세서 등 다양한 병렬처리 기술을 이용하였으며 처리 데이터의 크기를 조절할 수 있도록 구성하였다. 따라서 상대적으로 데이터의 동적 범위(Dynamic Range)가 작은 영상이나 음성 데이터를 처리하는 경우에 여러개의 데이터 처리기 구조로 분할되어 동시에 여러개의 데이터를 처리할 수 있어 성능을 개선시킬 수 있다. 그러나 이들은 다수의 기능 유니트들을 내장한 고가의 DSP 칩으로 die 면적을 많이 차지하고 전력소모가

많기 때문에 대부분의 고정 소수점 DSP 칩 응용영역, 특히 휴대용 단말기 영역에는 부적합하다.

따라서 본 논문에서는 무선 휴대 통신을 위한 저가·저전력이며 멀티미디어 실시간 서비스를 제공할 수 있는 새로운 고성능 고정 소수점 DSP 코어(Multimedia Fixed-point DSP Core : MDSP)를 개발한다. 새로운 명령어 집합 설계를 위해 기존의 멀티미디어 알고리즘을 분석하고 멀티미디어 확장 명령어^[5-12,16-20]를 분석하여 MDSP 명령어 집합 설계에 반영하였다. MDSP 명령어 집합은 멀티미디어 데이터의 병렬성을 이용하기 쉬운 SIMD 형태의 명령어, 전용의 DSP 명령어 및 특정 알고리즘 처리를 위한 벡터 명령어 등을 포함한다. 이를 지원하기 위해 MDSP 아키텍처는 SIMD, 벡터 프로세싱의 병렬처리 기술과 DSP 기술을 접목한 형태로 개발되었다. MDSP는 한 개의 데이터 경로가 목적에 따라 여러개로 분할될 때 8, 16, 32, 40 비트 등의 다양한 데이터 형태의 처리가 가능하며 또한 멀티미디어 응용영역에서 핵심적인 역할을 하는 MAC(Multiply and Accumulate) 연산을 한 사이클에 2개를 동시에 처리하여 성능을 향상시킨다.

MDSP는 4단계 명령어 파이프라인에 의한 클럭 속도 향상과 병렬처리로 인한 2개의 벡터 연산 및 2개의 데이터 이동을 동시에 처리한다. 특히 새롭게 제안된 스위칭(Switching) 네트워크와 Packing 네트워크는 MPEG(Moving Picture Expert Group) 디코딩, 인코딩, 콘볼루션 등의 알고리즘에서 연산과 데이터 변환을 중첩시켜 성능을 향상시킨다.

본 논문은 다음과 같이 구성된다. 2장에서는 MDSP의 명령어 집합에 대해 기술하며 3장에서는 새로운 MDSP 아키텍처에 대해 논한다. 4장에서는 이의 HDL(Hardware Description Language) 모델 설계 및 성능을 기술하며, 끝으로 5장에서 결론을 맺는다.

II. MDSP 명령어 집합

이 장에서는 MDSP의 명령어 집합 설계에 대해 기술한다. 멀티미디어 알고리즘과 고정 소수점 DSP 칩의 응용 분야에서 주로 사용되는 알고리즘들을 분석하고 기존의 고정 소수점 DSP 칩과 멀티미디어 명령어 집합^[1-12,16-20]을 분석하여 새로운 명령어 집합 설계에 반영하였다. 설계된 명령어 집합은 병렬처리 기술

을 이용하여 성능을 개선시키는 것으로 요약된다. 이러한 병렬처리는 SIMD, DSP, 벡터 프로세싱의 성질을 이용한 CPU(Central Processing Unit) 연산으로 요약된다. 따라서 제안한 명령어는 데이터 precision 이 작으며 지역적이고 반복적인 연산을 수행하는 알고리즘들에 유리하다. 이러한 알고리즘들은 MPEG 오디오 및 비디오, 영상처리, 디지털 필터 등을 예로 들 수 있다.

위의 알고리즘을 효율적으로 처리하고 SIMD 아키텍처의 병렬성을 이용하기 위해 MDSP는 데이터 타입을 3가지 형태 - 나눔 바이트(Partitioned Byte), 나눔 워드(Partitioned Word), 나눔 더블워드(Partitioned Double-word) - 로 구분하였다. 하나의 나눔 바이트는 4개의 바이트 데이터가 한개의 32 비트 워드에 저장된다. 이 데이터 타입은 서로 다른 비트로 구성되므로 일반적인 32 비트 워드와는 구별된다. 따라서 이 데이터 타입을 이용하면 미디어 데이터는 하나의 나눔 바이트에 1개 화소의 칼라 영상 정보 - 적색, 녹색, 청색, 및 alpha 값 - 를 저장할 수 있어서 유리하다.

MDSP 명령어 집합의 또다른 특징은 포화(Saturation) 연산에 있다. 포화 연산은 나눔 데이터가 연산되는 과정에서 표현되는 수의 허용 범위를 넘어서는 경우 연산의 결과를 최대 또는 최소값으로 채워준다. 예를 들어, 나눔 가산은 2개의 나눔 바이트를 가산하며 연산의 오버플로우(Overflow)가 발생하는 경우에 연산결과를 최대값으로 채워준다. 마찬가지로 언더플로우(Underflow)가 발생하는 나눔 감산의 경우에는 연산결과를 최소값으로 채워준다. 이런 포화 연산은 영상 합성, 음향 신호 혼합 등의 알고리즘에서 잦은 오버플로우 또는 언더플로우의 exception 처리 부담을 줄여 제어 하드웨어를 간단히 구현할 수 있게 해준다.

프로세서가 미디어 프로그램에 대해 성능을 향상시키기 위해서는 하나의 32 비트안에 포함된 여러 데이터에 대해 최대한 동시에 연산을 가능하도록 해야한다. 따라서 본 MDSP 멀티미디어 명령어 집합은 병렬성을 이용하기 쉬운 기본적인 산술 명령어(ADD, SUB, MUL), 논리 명령어(AND, OR, XOR, NEG), 비교 연산 명령어(MAX, MIN), 데이터 소자를 묶거나 풀어주는 변환 명령어(PACK, UNPACK), 쉬프트 명령어(ASL, ASR, LSL, LSR) 등을 기본으로 구성된다.

이외에도 MDSP에는 DSP가 갖는 전용의 MAC 명령어와 비트 필드 조작 명령어를 가지며 코어의 동작 모드와 명령어의 조합을 통한 벡터 명령어들을 제공한다. 이는 표 1에 나타내었다.

표 1에서 MDSP의 명령어 집합은 데이터의 처리 단위를 8, 16, 32, 40 비트로 분할하였다. 분할된 8, 16 비트 데이터 처리 단위는 Low-precision 또는 영상 처리에 주로 사용되며 분할된 32, 40 비트 데이터 처리 단위는 음성신호 및 고해상도 영상처리에 주로 사용된다.

비교 명령어는 복합적인 데이터에 대해서 동시에 여러개의 조건을 체크하거나 데이터 화소를 비교하여 값을 설정하는 알고리즘에 유리하다. 데이터 변환 명령어는 음성과 영상 등 여러 종류의 데이터가 혼합된 데이터에서 원하는 종류의 데이터만을 추출하거나, 연산 가능한 데이터 형태 또는 출력될 데이터로 전환하는데 효과적이다. 데이터 변환을 위하여 기본적으로 데이터를 묶거나 풀어주는 명령어(PACK, UNPACK) 이외에도 원하는 데이터 형태를 만들기 위한 특수 변환 명령어(PGCOPY, PSHF, PDIST)를 제공한다. PGCOPY는 2개의 32 비트 워드의 한 부분을 복사하는 명령어이며, PSHF는 2개의 32 비트 데이터를 바이트 단위로 shuffle시키는 명령어이며, PDIST는 32 비트 안에 분할된 4개의 바이트가 취할 수 있는 모든 순열에 대해서 데이터를 복사하는 명령어이다. 이를 이용하면 4 x 4 matrix 변환에 유리하다.

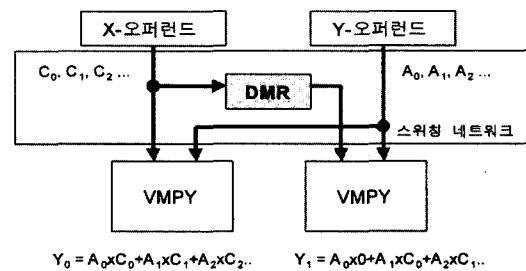


그림 1. 콘볼루션 연산
Fig. 1. Convolution Operation.

MAC 연산은 디지털 필터의 핵심으로 사실상 모든 멀티미디어 응용 영역에서 핵심적인 역할을 한다. 설계된 MDSP의 MAC 연산은 한 사이클에 4개의 8 x 8 곱셈 및 4개의 16 비트 누적 연산, 또는 2개의 16 x 16 곱셈 및 40 비트 누적연산을 동시에 처리한다. 또한 곱셈된 4개의 8 x 8 곱셈 결과를 모두 가산해서

packing 시켜주는 벡터 연산(PMAD, PMAD16)과 스위칭 네트워크의 지연 소자인 DMR(Double-MAC Register)을 이용하여 그림 1과 같이 한 사이클에 2개의 콘볼루션 결과를 얻을 수 있는 벡터연산(DMAC)을 지원한다. 또한 입력 데이터를 OMR 레지스터를 이용 다양하게 변환하여 여러 형태의 벡터 연산을 지원할 수 있어 유리하다.

표 1. 명령어 집합
Table 1. Instruction Set.

연산 종류	명령어 종류	데이터 처리 단위
가산 / 감산	PADD (wraparound) (signed/unsigned sat.)	8, 16, 32, 40 8, 16, 32, 40
	PSUB (wraparound) (signed/unsigned sat.)	8, 16, 32, 40 8, 16, 32, 40
	PADC, PSUC(add/sub with carry)	32, 40
	PINC, PDEC (increment, decrement)	32, 40
논리	PAND, POR, PXOR, PNEG	40
비교	PMAX, PMIN, SMAX*, SMIN* (parallel/serial min/max)	8
	Pack, Unpack	8, 16
변환	PGCOPY (group copy)	8
	PSHF (shuffle)	8
	PDIST (distribute)	8
쉬프트	ASL, ASR, LSL, LSR ROL, ROR	8, 16, 40 40
	곱셈 / MAC	PMUL (multiply)
PMAC* (multiply and accumulate)		8, 16 (4개의 8x8+16=16, 2개의 16x16+40=40)
PDMAC* (double MAC for convolution)		8, 16 (4개의 8x8+16=16, 2개의 16x16+40=40)
PMADB*, PMADB16* (multiply and add, packing)		8
SAD	PSAD*, PSAD16* (sum of absolute, packing)	8
제어	POR, FOREVER, REP, EXITcc (H/W 루프 및 반복 명령)	
	JMP, JMPcc, CALL, CALLcc, RTI, RTS (분기 명령)	
	WAIT, IDLE (저전력 명령)	
	TRAP, NOP (S/W interrupt)	

* 벡터 연산명령어

MDSP의 곱셈 명령어 집합은 하나의 상수 데이터(Constant)로 다른 데이터에 대해 분산하여 곱할 수 있다. 이러한 연산은 고정된 값으로 곱셈을 하는 많은 신호처리 및 영상처리 알고리즘에 도움을 준다. 그러나 같은 연산을 기존의 MMX^[14,17], MAX-2^[18]를 이용하여 처리하는 경우 메모리에서 상수값을 4번 복제한 후 곱셈을 수행해야 하므로 연산시 사이클이 많이 소모된다. MDSP만이 가지는 장점은 OMR (Operating Mode Register)과 명령어를 이용한 벡

터 연산에 있다. 예를 들어 MAC과 같이 연산이 계속 반복되는 경우와 특정 알고리즘의 반복 연산을 처리하기 위해 MDSP는 OMR 레지스터를 이용 처리 데이터를 벡터화 시킬수 있다. DCT 연산에 이 방법을 적용하면 2개의 데이터 이동, 4번의 데이터 복사, 4개의 8 x 8 곱셈, 4개의 데이터 누적 또는 4개의 데이터 가산, 및 2개의 데이터 packing이 한 사이클에 동시에 처리된다.

본 MDSP의 명령어 집합의 또다른 특징은 특수 응용 영역을 목표로한 전용 명령어를 수용한 점이다. 대표적으로, MPEG 부호화, 실시간 비디오 압축 및 패턴 인식에서 모션 측정을 위한 SAD(Sum of Absolute Difference) 연산 명령어(PSAD, PSAD16)는 4쌍의 8 비트 데이터의 절대값 차이에 의한 결과를 누산함으로써 12개의 연산을 한 사이클로 처리한다. 이외에도 MDSP는 DSP의 parallel move를 지원하므로써 동시에 연산과 2개의 데이터 이동을 중첩시키므로써 I/O bandwidth를 향상시킨다.

III. MDSP 아키텍처

MDSP는 프로그램 가능한 고정 소수점 DSP로 음성신호압축, 채널코딩 등 디지털 이동통신에서 사용하는 알고리즘 뿐 아니라 MPEG 디코딩, 인코딩 등 멀티미디어 알고리즘을 효율적으로 처리한다. MDSP는 그림 2와 같이 병렬로 동작하는 3개의 유니트로 구성된다. 이 유니트들은 전체 코어의 동작 제어와 명령어 페치 등을 관장하는 PCU(Program Control Unit), 데이터 메모리내의 오퍼랜드 위치를 계산하는 AGU (Address Generation Unit)와 연산부인 DPU (Data Processing Unit)로 구성된다.

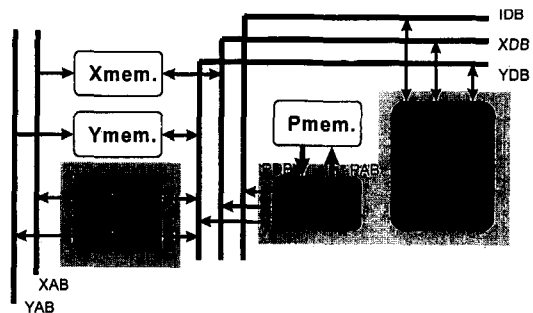


그림 2. MDSP 아키텍처
Fig. 2. MDSP Architecture.

MDSP는 한 사이클에 프로그램과 데이터를 이동시키기 위해 프로그램 메모리와 데이터 메모리를 별도로 갖는다. 프로그램 메모리는 내부에 2K x 24 비트를 가지며 최대 16M 워드까지 확장할 수 있다. 데이터 메모리는 한 사이클에 2개의 데이터를 이동시키기 위해 각각 X-메모리와 Y-메모리로 분리하여 구성하였다. 이들은 각각 내부에 2K x 32 비트를 가지며 XAGU, YAGU에서 계산된 주소로 액세스된다. 또한 각 X, Y-메모리는 최대 16M 워드까지 확장할 수 있다.

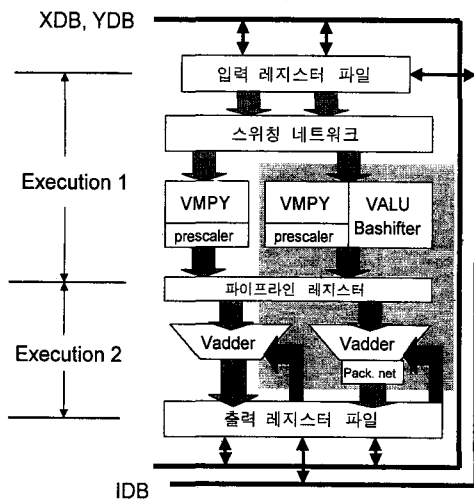


그림 3. DPU 아키텍처
Fig. 3. DPU Architecture.

그림 3의 DPU는 MDSP의 모든 산술 및 논리 연산을 수행한다. DPU는 파이프라인에 따라 2개의 execution 스테이지로 나뉘며 첫째 스테이지는 병렬로 동작하는 입력 레지스터 파일, 스위칭 네트워크, 2개의 벡터 곱셈기(Vector Multipliers : VMPY), 벡터 ALU(Vector ALU : VALU) 및 Barrel 쉬프터로 구성되며, 둘째 스테이지는 병렬로 동작하는 벡터 가산기(Vector Adder : Vadder)와 Packing 네트워크 및 출력 레지스터 파일로 구성된다.

두개의 벡터 곱셈기는 디지털 신호 처리부에서 자주 사용되며 고속을 요구하는 복잡한 신호처리 기능인 adaptive 필터, 상관 함수(Correlation Function), 및 FIR 필터를 효과적으로 처리하기 위한 목적과 영상 및 그래픽스 연산에서 matrix 곱셈을 가속하기 위해 제안되었다. 기존의 DSP 칩과는 달리 하나의 벡터 곱셈기는 그림 4와 같이 1개의 16 x 16 곱셈기와 2개

의 8 x 8의 곱셈기로 이루어져 있다. 따라서 동시에 1개의 16 x 16 또는 2개의 8 x 8 곱셈이 가능하다. 또한 벡터 가산기와 쌍을 이루어 1개의 16 x 16 또는 2개의 8 x 8 MAC 연산이 가능하다. 16 x 16 곱셈기에는 prescaler가 있어 소수점 연산, 정수 연산, 반복적인 MAC 연산시 overflow 방지를 위한 down-scaling 및 정밀한 연산을 위한 up-scaling을 지원한다.

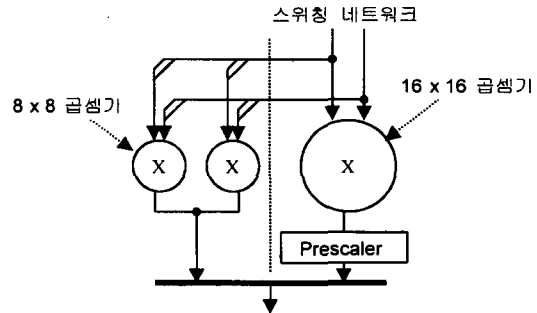


그림 4. 벡터 곱셈기의 구조
Fig. 4. Vector Multiplier Architecture.

DPU는 두개의 벡터 곱셈기와 두개의 벡터 가산기를 가지고 있어 2개의 벡터 MAC 구조로 이루어져 있다. 2개의 MAC 구조는 2개의 16 x 16 mixed 곱셈과 2개의 40 비트 가산, 또는 4개의 8 x 8 곱셈과 4개의 16 비트 가산을 한 사이클에 수행한다. 특히 그림 5(a)처럼 4개의 8 x 8 곱셈과 3개의 16 비트 가산을 이용하는 경우 한 사이클에 4 x 1 matrix 곱셈을 수행한다. 이는 4 x 4 matrix 형태로 표현되는 DCT, low-level 이미지, 그래픽스 등의 알고리즘 처리에 유효하다.

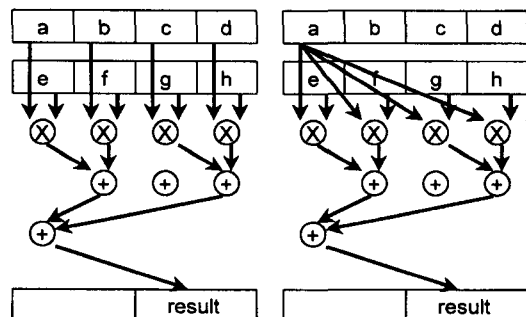


그림 5. (a) 4 x 1 Matrix 곱셈 (b) OMR을 이용한 상수 곱셈
Fig. 5. (a) 4 x 1 Matrix Multiplication (b) Constant Multiplication using OMR.

스위칭 네트워크는 MDSP의 동작 모드를 결정하는 OMR 레지스터에 의해 제어되며 벡터 곱셈기로의 입력 오퍼랜드를 다양한 형태로 변환시킬 수 있다. 따라서 특정 알고리즘 처리시 OMR을 연산의 목적에 맞게 설정하면 벡터 연산을 유도하게된다. 예를 들어 DCT 연산에서 한개의 계수가 데이터에 대해 분산하여 곱해질 때 다른 멀티미디어 DSP를 이용하는 경우 오퍼랜드를 미리 정렬해서 연산할 형태로 만들어 사용하게 된다. 그러나 MDSP는 사전 데이터 정렬을 하지 않고도 그림 5(b)처럼 상수 곱셈을 수행할 수 있다. 따라서 데이터 정렬을 위한 사이클을 절약할 수 있어서 성능을 향상시킨다. 스위칭 네트워크는 간단한 MUX로 구성되어 있다.

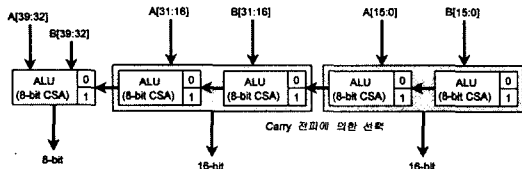


그림 6. 벡터 ALU의 구조
Fig. 6. Vector ALU Architecture.

벡터 ALU는 8, 16, 32, 40 비트 데이터에 대해서 산술 및 로직 연산을 수행한다. 벡터 ALU는 그림 6과 같이 5개의 8 비트 ALU로 구성되어 있으며 각각은 carry select 형태로 연결되어 있다. 8 비트 연산 시에는 각각의 8 비트 ALU가 독립적으로 연산을 수행하며 16 비트 연산시 2개가 쌍을 이루어 하위 8 비트에서 발생한 carry가 상위 8 비트로 인가되어 연산을 수행한다. 이러한 방식으로 32 비트, 40 비트 연산이 모두 가능하다. 또한 벡터 ALU는 벡터 가산기와 쌍을 이루어 벡터 연산을 수행한다. 이는 기존 DSP에 없는 구조로 2번의 ALU 결과를 이용할 수 있어 유리하다. 예를 들어 첫 번째 벡터 ALU에서 절대값 연산을 수행하고 두 번째 벡터 가산기에서 가산을 수행하면 한 사이클에 간단하게 SAD 값을 추출할 수 있다. MDSP는 벡터 곱셈기와 벡터 가산기를 쌍으로 또는 벡터 ALU와 벡터 가산기를 쌍으로 다양한 벡터 연산을 지원한다. MDSP에서 정의한 벡터 연산은 matrix 가산 및 감산, matrix 곱셈, MAC, 콘볼루션, SAD 등을 이용하는 알고리즘들에 특히 도움이 된다.

Packing 네트워크에서는 연산 결과가 16 비트만이 유효한 경우 전 데이터의 일부에 값을 저장시키므로써

레지스터 파일 및 메모리의 활용도를 높일 수 있다. Packing 네트워크의 동작 또한 연산과 중첩될수 있어 이를 이용하는 알고리즘인 경우 성능이 향상된다.

4개의 32 비트 워드로 구성된 입력 레지스터 파일은 연산의 source 오퍼랜드 또는 레지터 메모리로부터 연산기로의 데이터 이동을 위한 버퍼 역할을 한다. 4개의 40 비트 워드로 구성된 출력 레지스터 파일은 입력 레지스터 파일이 가지는 기능 이외에도 출력 값을 저장하며 어큐뮬레이터로 사용될 때는 기본 베이스 레지스터로 활용된다.

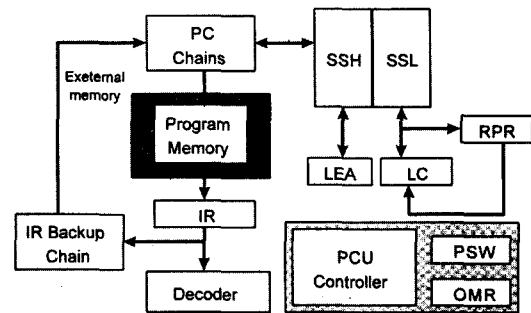


그림 7. PCU 아키텍처
Fig. 7. PCU Architecture.

그림 7의 PCU는 MDSP의 프로그램 제어를 관장하는 유니트로 명령어 페치, 디코딩, 하드웨어 루프 및 REP 명령어 제어, 각종 jump, subroutine call 등과 같은 명령어 제어 및 코어 내·외부의 인터럽트 서비스를 관리한다. PCU는 5개의 24 비트 레지스터 PC(Program Counter), IR(Instruction Register), PSW(Process Status Word), LEA(Loop End Address), OMR(Operating Mode Register), 한개의 16 비트 LC(Loop Counter) 및 시스템 스택을 갖는다.

PC는 프로그램 메모리로부터 페치한 다음 명령어의 주소를 갖고, IR 레지스터는 페치한 명령어를 저장하며 명령어 디코더로 이 값을 전송시키는 파이프라인 레지스터이다. PSW 레지스터는 DPU의 연산 상태와 하드웨어 루프 상태를 나타내며, LEA 레지스터는 루프 내의 마지막 명령어 위치를 나타낸다. LEA 값은 루프 시작 전에 루프의 nesting을 위해 스택에 저장되고 루프를 탈출할때 스택으로부터 복귀된다. OMR은 MDSP의 현 동작 모드를 반영하며 OMR을 참조하는 명령어에 의해서만 영향을 받는다. 16 비트 루프

카운터는 하드웨어 루프의 반복 회수를 계산하는 down counter이다.

시스템 스택은 16 레벨의 24 비트 SSH(System Stack High-word)와 SSL(System Stack Low-word)로 구성된다. 이것은 subroutine call과 인터럽트 발생동안 PC와 PSW를 저장하며 하드웨어 루프 처리를 위해 LEA와 LC 레지스터 값도 저장한다. 또한 한 개의 명령어를 반복 수행하는 경우 루프 명령어를 사용하면 스택 조작의 오버헤드와 연속된 루프의 개수를 감소시키며 프로그램 메모리 크기를 증가시키는 요인이 된다. 따라서 약간의 하드웨어를 이용 단일 명령어로 구성된 REP 루프를 지원하였다. 이를 위해 LC 값을 저장하기 위한 RPR(RePeat Register) 레지스터가 추가되었다.

AGU는 데이터 메모리에서 오퍼랜드를 지정하는데 필요한 주소 계산을 수행하며, 가산기들과 어드레스 레지스터들로 구성된다. 가능한 어드레싱 모드로는 레지스터 직접, 레지스터 간접, Immediate, Bit-reversed, 읍셋, 모듈로, 선형 모드가 있다. AGU는 XAGU와 YAGU로 구성되며 그림 8은 XAGU의 아키텍처를 나타낸다. XAGU는 Xmr0(XAGU Modifier Register), Xmr1 레지스터로 제어되며, YAGU는 Ymr0(YAGU Modifier Register), Ymr1 레지스터로 제어된다. 이 레지스터들은 상태에 따라 어드레싱 방법을 Bit-reversed, 읍셋, 모듈로, 선형 모드로 변경시킨다.

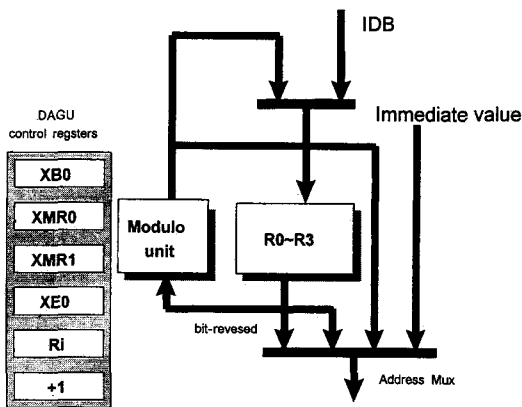


그림 8. XAGU 아키텍처
Fig. 8. XAGU Architecture.

XAGU의 읍셋 및 어드레스 레지스터는 Ri와 R0-R3이며, YAGU는 Rj와 R4-R7이다. Ri와 Rj는

읍셋 레지스터로써, XAGU와 YAGU에서 주소를 변경하는데 필요한 읍셋 값을 저장해준다. 본 DSP 칩은 사용자에게 프로그램 작성의 융통성(Versatility)과 설계된 마이크로 코드의 비트 수를 줄이기 위해 다양한 어드레싱 모드를 제공하며 또한 8개의 어드레스 레지스터를 두었다.

IV. 논리합성 및 성능평가

이 장에서는 MDSP의 논리합성 및 성능에 대해 기술한다. 설계한 MDSP는 MPW를 이용하여 칩으로 구현하였다. MPW에서 지원하는 최대 게이트 수가 8만 게이트로 제약이 되어 있기 때문에 PCU 및 AGU를 12 비트로 설계를 변경하였다. 결정된 설계 사양에 따라서 Verilog HDL로 모델링을 수행하고 Cadence^(TM) 캐드 툴을 이용하여 시뮬레이션 하였다. 논리 합성은 0.6 μm SOG 라이브러리(KG75000)를 이용하여 SYNOPSISTM 캐드 툴로 합성하였다. Fanin/out, rising/falling time 등과 같이 실제 회로상의 특성들이 고려된 전체 게이트수는 메모리를 제외하고 68,831개이며 worst case 동작 주파수는 SOG 라이브러리의 제약으로 인해 30 MHz이다. 표 2는 칩의 특성을 정리한 표이다.

표 2. MDSP 칩의 특성
Table 2. Characteristics of the MDSP Chip.

사양	MDSP
Technology	0.6 μm SOG
게이트 수	68,831
동작 주파수	30 MHz

구현된 칩은 테스트 보드를 제작하여 기능을 검증 중이며 현재 일부 알고리즘에 대한 검증은 완료된 상태이다. 그림 9(a)는 구현한 테스트 보드를 나타낸다. 테스트 보드는 PC Interface를 위한 8255 블럭, 8개의 8 비트 SRAM을 이용한 두개의 32 비트 데이터 메모리(X, Y), 세개의 SRAM을 이용하여 24 비트 프로그램 메모리 및 PC와 MDSP간의 메모리 제어를 위한 제어 회로 등으로 구성되어 있다. 테스트 보드는 두가지 동작 모드를 가지고 있다. Download 모드에서는 PC를 통해 프로그램을 프로그램 메모리로 download 시키며 Run 모드에서는 프로그램 메모리에 저장된 프로그램을 이용하여 MDSP를 구동시키는

모드로 Logic Analyzer를 이용하여 MDSP의 데이터 출력을 검증하였다. 시뮬레이션 상 칩은 30 MHz까지 동작하나 보드상에서는 와이어 래핑으로 인해 최대 20 MHz까지만 테스트가 가능하였다. 그림 9(b)는 DCT 알고리즘을 수행한 Logic Analyzer의 출력 파형이다.

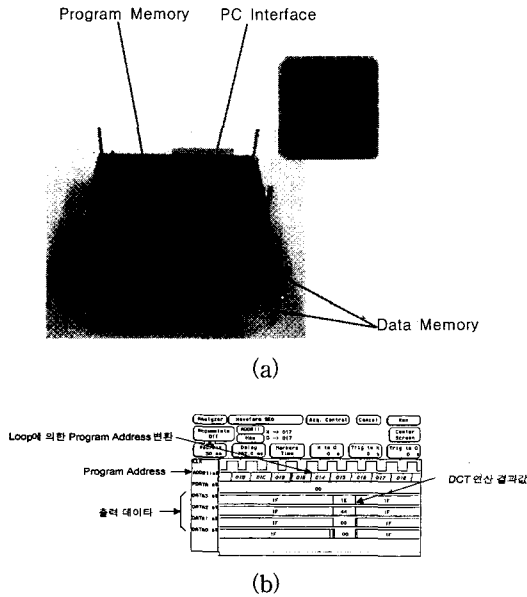


그림 9. (a) 테스트 보드 사진 및 (b) DCT 알고리즘 결과 파형
 Fig. 9. (a) Photograph of the Test Board and (b) Result Waveform of the DCT Algorithm

JPEG, MPEG 디코딩을 위해 DCT 프로그램을 벤치마킹하였다. 컬러 영상 8 x 8 블록에 대한 2차원 DCT 프로그램은 Chen W. H.의 알고리즘^[21]을 이용하였다. 움직임 추정은 MPEG의 인코딩 과정에서 가장 계산량이 많으며 움직임 벡터를 구하는 과정이다. 여기서 움직임 벡터를 구하는 여러 알고리즘 중 가장 일반적인 방법인 BMA(Block Matching Algorithm) 알고리즘을 사용하였다. 이외에도 MDSP는 신호처리 알고리즘인 FFT, IIR, FIR, Adaptive 필터 등에 대해 Motorola DSP56100 칩^[1], TI TMS320C5x 칩^[3]과 사이클 수만을 비교한 결과 표 3처럼 약 1~2 배 성능 향상을 얻는다.

기존의 고정 소수점 DSP 칩이 약 7 ~ 8만 게이트인 것을 고려할 때 설계한 MDSP는 간단한 RISC 형태의 명령어를 지원하고 있기 때문에 거의 같은 게이트 수를 가지며 동시에 2개의 MAC 연산이 가능하여 성능은 최대 2배까지 향상시켰다. 뿐만 아니라

SIMD, 벡터 프로세싱 등의 병렬 처리 기술의 사용으로 기존의 고정 소수점 DSP 칩으로는 처리가 어려운 멀티미디어 데이터 처리가 용이하다.

표 3. 기존 DSP 칩과의 성능 비교
 Table 3. Performance Comparisons with Existing DSP Chips.

기능	DSP56100	TMS320C5x	MDSP
FFT	10N	$9.5N + 32N^{1/2} + 4N^{1/3}$	7N
IIR	6N	12N	5N
FIR	N	N	N/2
Adaptive Filter	3N	3N	2N
DCT (720 x 480)	5 frames/sec		
BMA (352 x 240)	3 frames/sec		

V. 결론

기존의 고정 소수점 DSP 칩은 주로 신호처리 전용으로 멀티미디어 데이터 처리에 부적합하다. 또한 기존의 멀티미디어 DSP 칩은 수백만개 이상의 트랜지스터가 집적된 고가의 칩으로 die 면적을 많이 차지하고 전력소모가 많기 때문에 휴대용 응용영역에는 부적합하다. 따라서 본 논문에서는 휴대 멀티미디어 응용에 적합한 새로운 방식의 멀티미디어 고정 소수점 DSP(MDSP)를 제안하였다. MDSP는 휴대용 멀티미디어 응용 영역에 적합하도록 다수의 기능 유닛을 배제하고 두 개의 MAC 유닛과 새롭게 제안한 스위칭 네트워크 및 Packing 네트워크를 통한 벡터 연산을 이용하여 성능을 향상시켰다. 또한 멀티미디어 데이터 처리에 적합하도록 SIMD 기술을 도입하여 동시에 4개의 바이트 2개의 워드 연산이 가능하게 하였고 기존의 DSP 기술을 접목하여 1차원 신호처리 알고리즘에도 효율적이다. MDSP는 Verilog HDL을 이용하여 모델링 하였으며 0.6 μm SOG 라이브러리(KG75000)를 이용하여 SYNOPSIS™ 캐드 툴로 논리합성 하였다. 설계한 MDSP는 MPW를 이용하여 칩으로 구현하였으며 신호의 버퍼링을 고려한 전체 게이트 수는 68,831개이고 동작 주파수는 30 MHz이다.

참고 문헌

[1] Motorola Inc., *DSP56100 Digital Signal Processor User's Manual*, 1994.

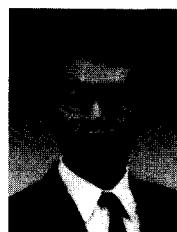
- [2] AT&T Microelectronics Inc., *DSP1610 Digital Signal Processor*, 1992.
- [3] Texas Instruments Inc., *TMS320C5x User's Guide*, 1990.
- [4] Analog Devices Inc., *ADSP2100 Family User's Manual*, 1993.
- [5] C. Hansen, "Architecture of a Broadband Mediaprocessor," *COMPCON 1996*, *IEEE Computer Society Press*, 1996.
- [6] C. Hansen, "MicroUnity's Media-Processor Architecture," *IEEE Micro*, vol. 16 no. 4, pp. 34-41, Aug. 1996.
- [7] Microprocessor Report, "Chromatic's Mpack2 Boosts 3D," pp. 5-10., Nov. 1996
- [8] Robert E. Owen, S. Purcell, "An Enhanced DSP Architecture for the Seven Multimedia Functions: The Mpack2 Media Processor," *IEEE Workshop on SiGNAL Processing Systems*, pp. 76-85, Nov. 1997.
- [9] Philips Semiconductors, "VLIW Computer Architecture," July 1996.
- [10] Ray Simar Jr., "DSP Architectures, Algorithms, and Code-Generation: Fission or Fusion?," *IEEE Workshop on SiGNAL Processing Systems*, pp. 50-59, Nov. 1997.
- [11] Texas Instruments Inc., *TMS320C62xx User's Manual*, 1997.
- [12] Analog Devices Inc., *ADSP 2106x SHARC User's Manual*, 1996.
- [13] Lavi A. Lev., Andy Charnas. et al., "A 64-b Microprocessor with Multimedia Support," *IEEE Journal of Solid-State Circuits*, vol. 30 no. 11, pp. 1227-1238, Nov. 1995.
- [14] Peleg A. and Weiser U., "MMX Technology Extension to the Intel Architecture," *IEEE Micro*, vol 16 no. 4, pp. 42-50, Aug. 1996.
- [15] Lee R., "Real-time MPEG Video via Software Decompression on a PA-RISC Processor," *Proceeding of IEEE COMPCON*, pp. 186-192. Mar. 1995.
- [16] Trembley M., O'Connor M., Narayannan V., He L., "VIS Speeds New Media Processing," *IEEE Micro*, vol. 16 no. 4, pp. 10-20, Aug. 1996.
- [17] Intel co., *Intel Architecture MMX™ Technology*, Mar. 1996.
- [18] Lee R., "Subword Parallelism with MAX2," *IEEE Micro*, vol. 16 no. 4, pp. 51-59, Aug. 1996.
- [19] Silicon Graphics Inc., "MIPS Extension for Digital Media with 3D," Dec. 1996.
- [20] Microprocessor Forum, Oct. 1996.
- [21] Chen W.H., Harrison S. C., and Fralic S. C., "A Fast Computational Algorithm for the Discrete Cosine Transform," *IEEE Trans. on Communications*, vol. COM-25, no. 9, sep. 1977.

저 자 소 개



尹成賢(正會員)

1996년 2월 아주대학교 전자공학 학사. 1996년 2월 ~ 1998년 2월 아주대학교 전자공학 석사. 1998년 3월 삼성 시스템 LSI 선형 연구팀. 주관 심분야는 DSP 칩, 통신 및 신호처리용 ASIC 설계



鮮于明勳(正會員)

1980년 2월 서강대학교 전자공학 학사. 1982년 2월 한국과학기술원 전자공학 석사. 1982년 3월 ~ 1985년 8월 한국전자통신연구소(ETRI) 연구원. 1985년 9월 ~ 1990년 8월 Univ. of Texas at Austin 전자공학 박사. 1990년 8월 ~ 1992년 8월 Motorola, DSP Chip Division, 미국. 1992년 8월 ~ 1996년 10월 아주대학교 전기전자공학부 조교수. 1996년 10월 ~ 현재 아주대학교 전기전자공학부 부교수