

## DSP 기능 유닛을 내장한 32 비트 RISC 마이크로프로세서의 구조 설계

안 상준, 정 우경, 김 문경, 문 상국, 이 용석

연세대학교 전자공학과

서울시 서대문구 신촌동 134 번지

Tel 02)361-2872, Fax 02)312-4584

asj@dubiki.yonsei.ac.kr

### The Architecture Design of 32-bit RISC Microprocessor with DSP Functional Unit

Sangjun An, Wookyeong Jeong, Moongyung Kim, Sangook Moon, Yongsuk Lee

Dept. of Elec. Eng., Yonsei University

134, Shinchon-dong, Seodaemun-gu, Seoul

Tel 02)361-2872, Fax 02)312-4584

asj@dubiki.yonsei.ac.kr

### 요약

본 논문에서는 내장형 응용에 적합한 RISC 마이크로프로세서와 DSP 프로세서의 기능을 유기적으로 결합한 구조를 연구하고 이를 설계한다. 프로그램의 크기를 줄이기 위해 RISC 명령어는 16 비트 명령어 집합을 설계하고 분기 명령어로 인한 손실을 줄이기 위해 한 개의 자연 슬롯을 갖고 있다. DSP 명령어는 32 비트 길이를 갖고 한 명령어로 곱셈, 덧셈(뺄셈), 두 가지 데이터 이동을 할 수 있어서 한 사이클에 최대 네 가지 동작을 할 수 있다. 파이프라인 단계는 IF, ID, EX, MA, WB/DSP 의 다섯 단계로 구성된다. DSP 기능을 지원하기 위해 내부 루프 버퍼를 갖고 정수 실행부에서는 주소 발생을 위한 전용 하드웨어와 DSP 유닛에서는 곱셈 및 누적 기능을 지원하기 위한  $17 \times 17$  비트 곱셈기를 내장된다. 제안된 구조의 설계는 Verilog-HDL을 이용하여 top-down 설계 방식으로 설계되었고 각 기능 검증을 마친 후  $3.3V$ ,  $0.6\mu m$  CMOS triple metal single poly 공정을 이용하여 합성하고 테이아웃 하였다.

### 1. 서 론

내장형으로 사용되는 CPU는 대부분이 32-비트 RISC (Reduced Instruction Set Computer) 구조를 채택하고 있다. RISC는 대부분의 명령어를 단일 사이클 내에 수행할 수 있는 특징이 있는데 최근의 무선 랜, 고성능 레이저 프린터, 휴대용 단말기, 네트워크 브리지와 라우터 등과 같이 제어 기능과 DSP 신호 처리 능력을 요구하는 응용 시장에서는 그 성능이 부족하다. 이와 관

※ 본 연구의 일부는 과학 재단의 특정 기초 연구 과제 연구비 지원으로 이루어 졌음. (97-0100-0701-2)

련된 최근의 연구<sup>[1]</sup>는 내장형 프로세서에 코프로세서 형태로 DSP 프로세서를 추가하는 방법이 있는데 이는 성능상의 한계와 프로그래밍이 어려워지는 단점이 있다. 본 연구에서는 이 DSP 프로세서의 기능을 할 수 있는 DSP 기능 유닛을 RISC 코어에 내장하고 RISC 코어에서는 이를 지원하기 위한 최적의 구조를 제시한다.<sup>[2]</sup>

각종 DSP 응용의 핵심은 곱셈 및 이의 누적 기능인데 이를 빨리 처리하기 위해 본 논문에서는 단일 사이클에  $17 \times 17$  비트 곱셈을 처리하는 곱셈기와 그 결과를 누적하는 누적기<sup>[3]</sup>를 DSP 기능 유닛에 첨가하였고 RISC 명령어의 정수 곱셈도 이 하드웨어를 공유하여 사용하도록 한다.

본 논문의 구성은 다음과 같다. 제 2 장에서는 프로세서 전체 구조에 대해 설명하고, 제 3 장에서는 설계된 회로의 검증 및 성능 평가를 하고, 제 4 장에서 결론을 맺는다.

### 2. YS-RDSP 프로세서의 구조

본 논문의 명령어 집합은 내장형으로 사용될 목적而已 때문에 응용 프로그램의 크기를 줄이기 위해서 데이터 패스는 32 비트이지만 명령어 길이는 16 비트 RISC 명령어 집합과 32 비트 DSP 명령어 집합으로 구성되어 있다. DSP 명령어는 디지털 신호 처리 알고리즘에서 가장 많이 사용되는 다음 수식을 처리하기에 적합한 구조로 32 비트 길이를 갖는다.

$$y(n) = \sum_{i=0}^{N-1} Input(n-i) Coeff(i) \quad (0 \leq n \leq N-1)$$

DSP 명령의 예 : ADD src1, src2, des1 MUL src3, src4, des2  
MOVX @Rm, des3 MOVY @Rn, des4

DSP 명령어는 조건부 실행 필드가 있는 명령어와 없는 명령어로 구분되며 조건부 실행 필드가 있으면 그 조건이 만족될 때 명령어가 실행되고 그 연산결과의 플래그는 갱신되지 않는다. 이것을 이용하면 분기 명령어는 사용되지 않고 프로그램 할 수 있고 이로 인한 성능상의 이점을 얻을 수 있다. RISC 명령어 집합은 C 언어에서 쓰이는 자동 포인터 증가 또는 감소와 반복된 루프에 해당하는 명령어를 포함하고 있어서 컴파일이 쉽게 되도록 지원한다. 또한 프로세서 상태를 과워다운 모드로 만드는 Sleep 명령과 입력 소스 중에서 큰 값이나 작은 값을 선택해서 저장하는 특수 명령어를 포함한다. 분기 명령어는 한 개의 지연 슬롯을 가지고 있어서 분기가 이루어진 분기 명령어의 실행은 한 사이클의 손실로 해결된다.

YS-RDSP 프로세서는 그림 1 과 같은 블록을 포함한다. RISC 명령어는 프로세서 제어 유닛의 제어 신호에 따라 프로그램 품에서 읽혀서 디코딩을 거쳐 정수 실행부에서 연산된 후 결과가 정수 레지스터 파일로 기록된다. DSP 명령어는 마찬가지로 디코딩을 거쳐서 DSP 유닛에서 실행된 후 DSP 레지스터 파일에 기록된다. 정수 실행부는 기본적인 ALU 와 주소 계산을 위한 전용 덧셈기인 AGU 를 둘이 있다. 이 유닛에서 발생된 주소는 Y 메모리 공간을 액세스하는데 사용되고 ALU 에서 발생시킨 주소로 X 메모리 공간을 액세스하는데 사용된다.

정수 실행부에는 32 비트 길이의 레지스터 16 개가 있다. R0 부터 R14 는 일반 용도의 레지스터이고 R15 는 스택(stack) 포인터(pointer)에 해당한다. 이들 레지스터 중 일부는 DSP 연산을 위한 주소 계산용으로 사용된다. DSP 유닛에는 6 개의 32 비트 레지스터와 2 개의 40 비트 레지스터가 있다. 40 비트 레지스터는 상위 8 비트가 가드(guard) 비트에 해당하며 하위 32 비트와 별도로 읽고 쓰기가 가능한 구조로 되어 있다. 정수 실행부 레지스터와 DSP 유닛 레지스터 사이의 데이터 전송은 RISC 명령어를 통해서 단일 사이클에 전달된다.

DSP 유닛은 16 비트 고정 소수점 데이터 형태를 갖지만 데이터 패스는 16 비트, 32 비트, 40 비트의 길이를 지원한다. 40 비트 데이터 패스는 소스가 가드 비트를 포함한 레지스터이면 그대로 입력되고 소스가 가드 비트가 없는 32 비트 레지스터이면 부호가 확장되어 40 비트로 만들어진다. 곱셈기는 부호 연산과 무부호 연산을 모두 지원하기 위해  $17 \times 17$  비트 입력을 갖고 33 비트 출력을 갖는 구조로 하였다. 곱셈 알고리즘은 radix-4 수정된 부스 알고리즘을 사용하여 9 개의 부분 곱을 더해서 한 사이클에 곱셈 연산을 종료하도록 설계하였다. 가변길이의 쉬프트 기능을 제공하기 위해 배럴 쉬프트를 갖고 있는데 산술 연산인 경우 쉬프트의 양은 +32 ~ -32 까지 가능하고 논리 연산인 경우 쉬프트의 양은 +16 ~ -16 까지 가능하다. 양의 값은 왼쪽 방향의 쉬프트를 의미하고 음의 값은 오른쪽 방향의 쉬프트를 의미한다.

YS-RDSP 프로세서는 DSP 연산을 위한 내부 메모리를 포함하고 있고 이 메모리는 프로세서의 전체 메

모리 맵의 일부분을 구성하고 있다. 크게 X 메모리와 Y 메모리로 구성되어 있으며 각 메모리는 4Kbytes 의 품과 4Kbytes 의 램이 있어서 총 8Kbytes 의 품과 8Kbytes 의 램 크기를 갖는다. 내부 버스는 I-bus, X-bus, Y-bus 로 3 가지가 있다. I-bus 는 32 비트 주소 버스와 32 비트 데이터 버스로 되어 있고 내부 메모리뿐만 아니라 외부 메모리 공간 모두를 액세스하기 위한 버스이다. X-bus 와 Y-bus 는 16 비트 주소 버스와 16 비트 데이터 버스로 되어 있고 DSP 연산을 위해 존재하며 각각 X 메모리와 Y 메모리, 즉 칩 내부 메모리 영역만 액세스 가능하다. 그래서 DSP 연산을 할 때 메모리 뱅크 충돌만 없다면 I-bus 를 통해서 명령어를 가져오고 X-bus 와 Y-bus 를 통해서 데이터를 가져오는 동작을 동시에 할 수 있다.

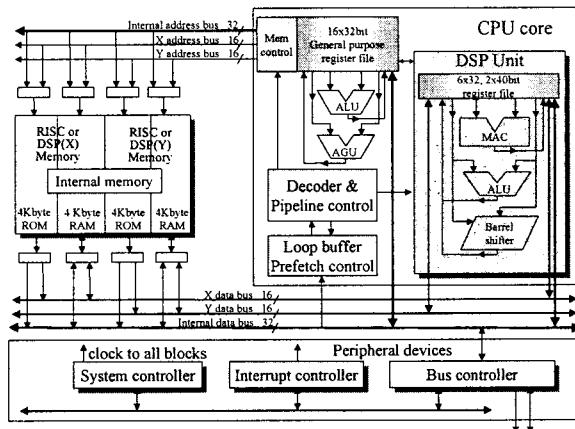


그림 1. YS-RDSP 프로세서의 전체 구조

주변장치로 YS-RDSP 프로세서는 시스템 제어기, 인터럽트 제어기, 버스 제어기를 갖고 있다. 각 주변장치에는 그 주변장치를 제어할 수 있는 레지스터가 있는데 이 레지스터들도 YS-RDSP 프로세서의 주소 공간의 일부로 지정되어 있고 발생된 주소로 버스 제어기를 통해 P-bus 로 연결이 되고 이들 레지스터를 읽고 쓸 수 있다. 시스템 제어기는 각 유닛으로 분배되는 클럭을 발생시키며 시스템 제어기 내에 있는 레지스터의 제어에 따라 칩의 동작 주파수를 결정한다. 저전력을 위한 파워다운(power-down) 모드는 3 가지를 지원한다. SLEEP 명령어의 실행을 통해서 프로세서는 내부의 제어 비트에 따라 CPU 코어만 멈추게 되는 sleep 모드와 CPU 코어와 주변장치 모두 멈추게 되는 대기 모드(standby mode)로 들어가게 되며 정상적인 동작 중에도 그 프로그램의 실행에 필요 없는 주변장치를 잠시 멈추게 하는 모듈 대기 모드(module standby mode)가 있다. 인터럽트 제어기는 외부나 주변장치에서 발생한 인터럽트의 우선순위를 정하고 그것에 따라 CPU 코어로 인터럽트 처리 요청을 보내는 일을 한다. 인터럽트 우선순위는 16 레벨을 지원하며 외부에서 들어온 NMI(Non-Maskable Interrupt)는 자동으로 우선순위 16 을 갖게 된다. 버스 제어기는 외부 메모리에 대한 인터페이스를 담당한다. 외부 메모리로 SRAM, DRAM, pseudo-SRAM, burst ROM 등을 지원하고 각 메모리에서

필요한 신호를 만들어서 별도의 제어 칩이 필요 없이 직접 연결이 가능하도록 하였다. 또한 외부 메모리의 액세스 속도 차이로 인한 대기 사이클(wait cycles)은 버스 제어기의 내부 레지스터의 프로그래밍으로 하거나 외부 입력된 대기 신호(WAIT 신호)에 의해 제어가 가능하다.

명령어를 처리하기 위한 파이프라인은 다음 그림 2 와 같다. 간단한 명령어는 (a)와 같이 IF(Instruction Fetch), ID(Instruction Decoding), EX(Execute) 3 단으로 종료하고 메모리 액세스나 DSP 명령어나 곱셈 명령어 같은 경우는 (b)와 같이 동작한다. 특히 정수 곱셈에서 통워드 곱셈인 경우 4 사이클이 소요되므로 바로 다음 명령어가 이 곱셈 하드웨어를 사용하면 파이프라인이 스톤된다.

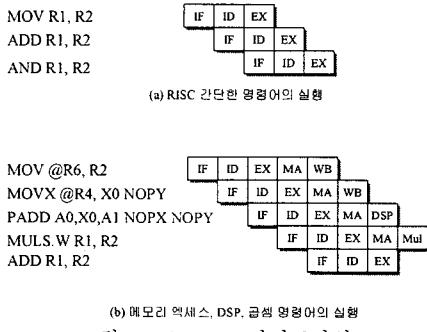


그림 2. YS-RDSP 파이프라인

본 논문에서 사용하는 내부 루프 버퍼는 보통 DSP 프로세서에서 사용하는 repeat buffer 형태에 캐쉬 기능을 갖도록 하였다. 즉 루프 버퍼에 루프 시작과 끝을 가리키는 BSC(buffer start counter)와 BEC(buffer end counter)를 두어 이 사이로 분기가 이루어지는 경우는 버퍼를 무효화시키지 않고 그대로 사용할 수 있도록 하였다. 이것은 최근에 실행된 명령어 n 개(n : 버퍼의 크기에 해당하는 명령어 수)를 무조건 버퍼에 담아 두었다가 유효한 영역에서 PC가 가리키는 명령어를 보내주게 된다. 루프 버퍼는 8 개 워드씩 4 개의 영역으로 구분된다. PC는 명령어를 읽을 위치를 가리키며 명령어 길이 만큼 4 또는 2 씩 증가되며, BSC는 유효한 명령어의 시작 위치를 가리켜서 BEC가 버퍼를 다 채우고 BSC로 돌아오면 밀려서 4 씩 증가한다. BEC는 현재 그 버퍼위치가 비어 있어서 프리페치하여 채워야 할 곳을 가리키며 4 씩 증가한다. BEC는 항상 4 씩 증가하므로 PC와 사이가 벌어지게 되며 그 차이가 4 개로 구분된 영역에서 한 영역이상 즉 8 워드 이상이 되면 프리페치가 멈추게 된다.

그림 3은 모듈로 주소 발생이 일어나는 데이터패스를 설명한다. 내부 메모리를 액세스하는 이중 데이터 전송 명령어가 실행되고 주소 발생은 사후 증가 모드가 되어 현재 레지스터 값이 주소로 사용되고 증가된 값이 그 레지스터를 변경시키는데 만약 주소가 모듈로 끝 주소와 일치하면 증가된 값이 아니라 모듈로 시작 주소가 그 레지스터에 저장된다. 모듈로 주소 발생은 DSP 알고리즘을 살펴보면 한 쪽만 지원해도 충분하기 때문에 하드웨어를 줄이기 위해 상태 레지스터

의 플래그가 가리키는 X나 Y 메모리 한 쪽만 지원한다.

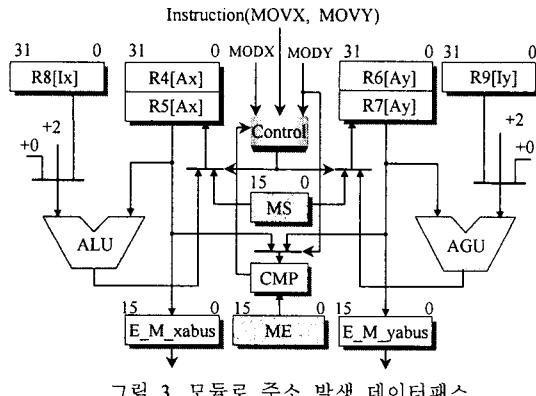


그림 3. 모듈로 주소 발생 데이터패스

DSP 연산은 32 비트 고정 소수점 형태로 연산되지만 메모리에 저장할 때 상위 16 비트로 된다. 이 때 하위 16 비트를 그대로 버리면 DSP 연산의 반복적인 특성으로 정확도가 많이 떨어진다. 따라서 이를 반올림하기 위한 명령어를 지원하는데 그 동작은 다음 그림 4와 같다.

어떤 수가 x.100...00 일 때 이것을 어떻게 처리하는가에 따라 성능이 달라지는데 하드웨어를 간단히 하기 위해 무조건 1의 위치에 1을 더해주는 방법과 부동 소수점 연산을 위한 IEEE 표준<sup>[4]</sup>에서 사용하는 Round-to-Nearest 방법과 양수일 때 반올림하고 음수일 때 하지 않는 방법 등 여러 가지가 있다. 본 논문에서는 Round-to-Nearest Even과 유사한 그림 4와 같은 방법으로 구현하여 성능을 높일 수 있었다.

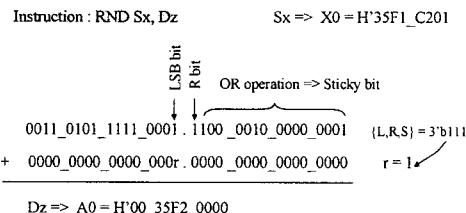


그림 4. 반올림 연산 방법

DSP 유닛의 곱셈기 블록 데이터 패스와 제어 블록은 그림 5와 같다. 곱셈기 제어는 곱셈기의 필요한 입력을 선택하고 결과를 저장하며 누적 연산을 위한 64 비트 덧셈기의 입력을 선택하고 MACH, MACL 레지스터의 입력과 쓰기 제어 신호를 만들어낸다. 곱셈이 부호와 무부호 연산을 동시에 지원해야 하기 때문에 곱셈기 입력으로 소수 16 비트에 부호 비트를 추가하여 17 비트 부호 곱셈을 하게 되면 곱셈기 설계가 간단해진다. 이 부호 비트는 명령어의 종류와 입력소스의 위치(상위 워드 또는 하위 워드)에 따라 결정된다. 곱셈기 자체는 17 x 17 곱셈을 하며 결과는 34 비트가 발생하지만 부호 비트의 중복으로 인해 이중 하위 33 비트가 실제 곱셈 결과가 된다.

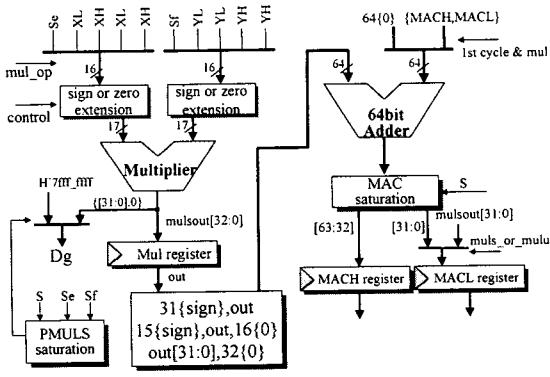


그림 5. 곱셈기 블록의 데이터패스

### 3. 성능 평가 및 설계 검증

본 논문에서 설계한 YS-RDSP 프로세서는 top-down 설계 방법에 따라 Verilog-HDL<sup>[5]</sup>을 이용하여 설계되었다. 검증은 먼저 각 기능 블록의 입출력 신호를 가지고 테스트 벡터를 만들어 검증하고 전체를 통합한 후 개별 명령어를 가지고 검증한다. 그 후 실제 사용되는 응용 프로그램을 가지고 검증 및 성능 평가를 하는데 이는 정수 응용 프로그램과 DSP 응용 프로그램으로 구성된다. 표 1은 DSP 응용 프로그램 실행 결과와 다른 DSP 프로세서의 실행 결과를 비교한 것이다.

표 1. DSP 응용 프로그램 실행 결과(단위 : 사이클 수)

프로그램명	OakDSP	TI C54x	YS-RDSP
Vecmultiply	620	758	488
Lattice Filter	1489	1768	634
Codebook	441	313	323
IIR Filter	1181	1764	729
FIR Filter	3068	5808	2930
FLRnoLoad	5868	13412	3729
JPEGDCT	3821	3787	3086

위의 검증은 설계의 기능 검증을 한 것이고 그림 6과 같은 과정에 따라 실제 게이트로 합성을 한 후 타이밍 검증을 해야 한다. 합성에 사용된 셀은 LG 반도체 3.3V, 0.6um CMOS 표준 셀이고 합성 툴은 SYNOPSYS 사와 COMPASS 사의 합성 툴을 사용하였고 타이밍 검증은 COMPASSQSIM을 이용하였다. 표 2는 설계된 YS-RDSP의 특성을 나타낸다.

표 2. YS-RDSP의 동작 특성

Process	3.3V, 0.6um TLM CMOS
게이트 수	113291 개(메모리 제외)
동작 주파수	42MHz
코어의 최대 Power 소모	538mW
코어의 크기	10.7mm X 8.4mm

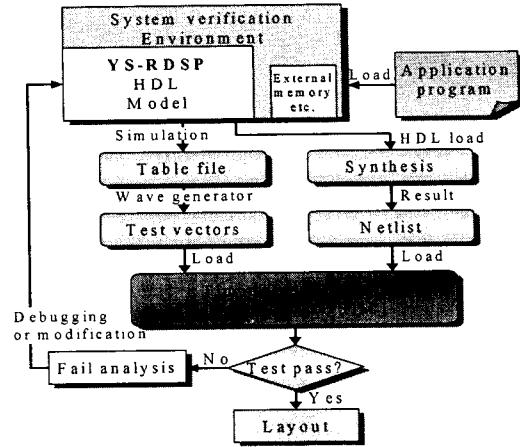


그림 6. 설계 검증 과정

### 4. 결 론

본 논문에서는 DSP 기능을 갖고 내장형 응용에 적합한 YS-RDSP 프로세서의 구조 및 필요한 기능에 대한 연구와 설계를 수행하였다. DSP 알고리즘에 적합한 RISC 코어의 구조와 DSP 명령어의 채택으로 DSP 프로그램 실행시 우수한 성능을 보였고 정수 프로그램으로 컴파일했을 때 그 프로그램 크기가 작아지는 것을 확인하였다. 시스템 제어기를 내장하여 칩의 동작 주파수와 각 기능 블록으로 공급되는 클럭의 제한으로 저전력 소비를 얻을 수 있었다.

본 논문에서 설계된 프로세서는 DSP 기능이 내장되어 있고 특히 전력 감소 모드를 다양하게 지원하기 때문에 저전력을 요구하는 이동 통신용 휴대형 단말기에서 그 성능을 강력하게 발휘할 수 있고 무선 통신용 CODEC 칩과 같이 사용하여 더 많은 응용 시장에 사용될 수 있다.

### 5. 참 고 문 헌

- [1] James Turley, *Selection a High-performance Embedded Microprocessor*, 2<sup>nd</sup> edition, MicroDesign Resources, pp. 189-218, 1997.
- [2] *Buyer's Guide to DSP Processors*, Berkeley Design Technology Inc., 1994.
- [3] 이용석, “곱셈기의 구조 및 설계”, 고성능 마이크로 프로세서 구조와 설계 강좌 시리즈, 1998, <http://mpu.yonsei.ac.kr/yslee/multiplier.html>
- [4] IEEE Standard for Binary Floating-Point Arithmetic, ANSI/IEEE Standard 754-1985.
- [5] IEEE Std. 1364-1995, *Hardware Description Language Based on the Verilog HDL*, Oct. 1996.