DSP기능을 강화한 RISC 프로세서 core의 ASIC 설계 연구

준회원 김 문 경*, 정 우 경*, 정회원 이 용 석*, 이 광 엽**

A Study on the Design of a RISC core with DSP Support

Moon-Gyung Kim*, Woo-Kyeong Jeong* *Associate Members,*

Yong-Surk Lee*, and Kwang-Youb Lee** *Regular Members*

요 약

본 논문에서는 RISC 마이크로프로세서에 DSP프로세서를 추가하여 멀티미디어 기능이 강화된 응용에 알맞은 마이크로프로세서(YS-RDSP)를 제안한다. YS-RDSP는 최대 4개의 명령어를 동시에 병렬로 처리할 수 있다. 프로그램의 크기를 줄이기 위해 YS-RDSP는 16비트와 32비트의 두 가지 명령어 길이를 지원한다. YS-RDSP는 칩 하나로 RISC마이크로프로세서의 programmability 및 제어능력에 DSP의 처리능력을 제공하기 위하여 8-KByte ROM과 8-KByte RAM을 내장하고 있다. 칩 내에 있는 주변장치중 하나인 시스템 컨트롤러는 저전압 동작을 위한 3가지의 전압강하모드를 지원하며 SLEEP명령어는 CPU코어와 주변장치의 동작상태를 변환시킨다. YS-RDSP프로세서는 Verilog-HDL를 이용하여 하향식설계방식으로 구현되었고 C-언어로 작성된 사이클 단위 시뮬레이터를 이용하여 개선되고 검증되었다. 검증된 모델은 0.6um, 3.3V CMOS 표준 셀 라이브러리로 합성되었으며 자동화 P&R에 의해 10.7mm*8.4mm코어 면적을 갖도록 레이아웃 되었다.

ABSTRACT

This paper proposed embedded application-specific microprocessor(YS-RDSP) whose structure has an additional DSP processor on chip. The YS-RDSP can execute maximum four instructions in parallel. To make program size shorter, 16-bit and 32-bit instruction lengths are supported in YS-RDSP. The YS-RDSP provides programmability, controllability, DSP processing ability, and includes eight-kilobyte on-chip ROM and eight-kilobyte RAM. System controller on the chip gives three power-down modes for low-power operation, and SLEEP instruction changes operation statue of CPU core and peripherals. YS-RDSP processor was implemented with Verilog HDL on top-down methodology, and it was improved and verified by cycle-based simulator written in C-language. The verified model was synthesized with 0.6um, 3.3V CMOS standard cell library, and the layout size was 10.7mm*8.4mm which was implemented by using automatic P&R software.

I. 서 론

최근 고성능 마이크로프로세서는 슈퍼스칼라 구조나 VLIW구조가 많이 채용되고 있으나 대규모 칩 면적과 높은 복잡도, 많은 전력소모로 인해 내장형 응용에는 적당하지 않다. 따라서, 내장형 시스템에 탑재되는 마이크로프로세서는 하나의 명령어를 한 사이클에 수행하는 스칼라 RISC구조가 효율적이다. 또한, 최근 마이크로프로세서는 휴대성과 통신 확대의 필요성에 따라 디지털 신호처리 능력이 요구되고 있다[1][2]. 일반 RISC제어기는 무선 LAN, 고성능 레이저 프린터, 이동전화기, 네트워크 브리지 및

* 연세대학교 전기전자공학과 Processor 연구실 (bungae@yonsei.ac.kr) ** 서경대학교 컴퓨터공학과 (kylee@skuniv.ac.kr)

논문번호 : K01135-0518, 접수일자 : 2001년 5월 18일

※ 본 연구는 1997년도 한국 학술진흥재단 대학부설연구소 과제 연구비에 의하여 연구되었음.

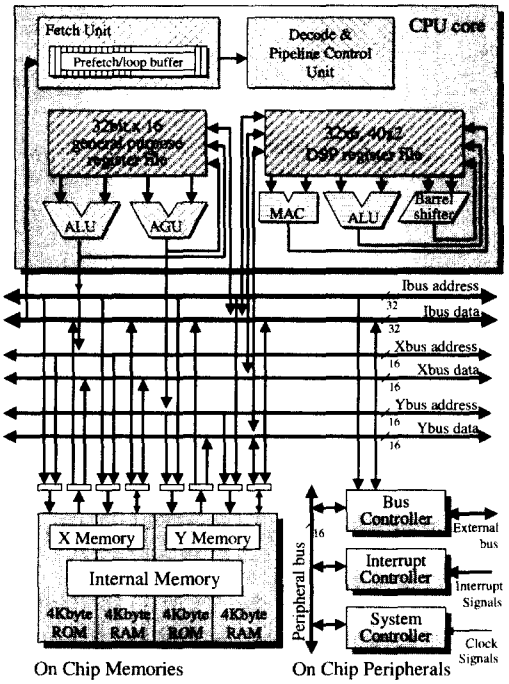


그림 1. YS-RDSP의 블록 다이어그램

라우터 같은 디지털 신호처리 능력뿐만 아니라 제어능력을 필요로 하는 응용 분야에는 성능과 유용성에서 한계를 갖는다.

대개 이러한 응용들에는 RISC제어기에 추가적인 DSP칩을 필요로 한다[3]. 코프로세서로 쓰이는 DSP 프로세서를 탑재하는 경향에 대한 최근의 연구들은 성능의 한계를 가져올 뿐 아니라 programmability 저하로 인한 단점을 갖는다[4]. 본 연구에서는 RISC와 DSP구조 각각의 특성과 장점을 조사하고, 이러한 특성을 최대한으로 살릴 수 있는 조합된 구조를 제안한다.

본 논문의 구성은 다음과 같다. 2장에서는 전체 프로세서의 구조에 대해, 3장에서는 본 프로세서의 동작을 테스트하기 위한 시뮬레이터의 설계 방법에 대해, 4장에서는 설계된 구조의 검증과 테스트에 대해 설명하고 5장에서 결론을 짓는다.

Ⅱ. YS-RDSP의 구조

YS-RDSP의 블록다이어그램은 그림 1과 같다. 명령어들은 ROM으로부터 페치유닛으로 읽혀지고 디코드유닛으로 보내진다. RISC ALU와 AGU는 32비트 정수 연산을 수행하고 이의 연산의 대상과 결

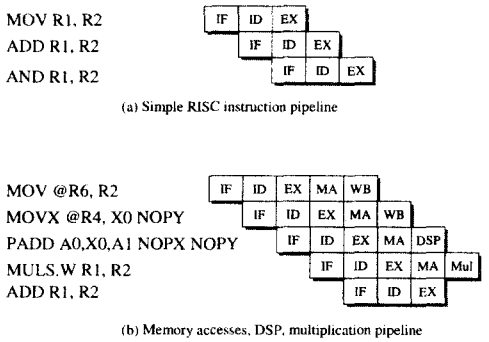


그림 2. YS-RDSP의 파이프라인

과는 일반 레지스터 파일에 저장되어 있다. AGU는 주소연산을 위한 덧셈기이다. DSP유닛은 MAC, ALU, 배럴쉬프터를 포함하고 있다. DSP연산의 대상데이터는 DSP레지스터 파일에서 읽혀지며 그 결과도 마찬가지로 그곳에 기록된다. DSP레지스터 파일은 2개의 40비트 레지스터와 6개의 32비트 레지스터로 이루어져 있다.

디코드/파이프라인 제어 유닛은 명령어를 해석하고 5단계 파이프라인을 제어한다. DSP동작 유닛은 다섯 번째 단계에서 DSP명령어를 수행한다. 간단한 RISC ALU명령어들은 세 번째 단계에서 종료된다. 그림 2는 YS-RDSP의 파이프라인 단계수행을 보여주고 있다. 다양한 흐름제어와 분기명령어가 제공되고 지연분기 같은 효율적인 분기명령어도 지원된다. 명령어 길이는 16비트 또는 32비트이다. 요구되는 프로그램 메모리를 줄이기 위해 RISC동작을 위해 32비트 데이터를 이용하는 짧은 16비트 명령어를 사용한다. YS-RDSP는 DSP기능을 수행하면서 높은 성능을 내기 위해 최대 2개의 DSP명령을 2개의 RISC명령과 병렬로 수행할 수 있는데, 대신 이런 복잡한 명령은 32비트 명령어를 필요로 한다. 대부분의 DSP알고리즘은 다음 형태의 연산을 포함한다.

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

inst.: PADD src1, src2, des1 PMUL src3, src4, des2
MOVX @Rm, des3 MOVY @Rn, des4

위와 같은 명령어는 한 사이클 내에 곱셈과 누적 덧셈, 두 주소변수 값의 자동 증가/감소를 수행할 수 있다. 단지 하나의 명령어를 데이터 수만큼 반복함으로써 DSP알고리즘을 완료시킬 수 있다.

루프들을 효과적으로 수행시키기 위해서 하드웨

어적인 루프동작 지원은 필수적이다. YS-RDSP는 루프를 지원하기 위해 페치유닛에 3개의 특수 레지스터를 가지고 있으며 각각은 RS(Repeat Start), RE(Repeat End), RC(Repeat Count)라 명명되어 있다. 프로그램의 흐름에 따라 PC값이 2또는 4로 증가되는데, 이 값이 RE와 같게 되면 RS값이 PC에 기록되고 RC값이 하나 줄어들게 된다. RC값이 0이 되면 루프수행이 종료되고 흐름상의 다음 명령어를 수행해 주게 된다. 루프를 수행해 주는 동안 명령어들은 프로그램 메모리에서 읽혀지지 않고 루프버퍼로부터 읽혀지게 된다. 이는 버스를 통한 메모리 제어로 기인한 스톨을 막을 수 있다. 루프버퍼는 16비트 32워드를 가지고 있고 프리페치 버퍼로도 사용된다.

Programmability와 제어능력을 향상시키기 위해, 몇몇 명령어들은 상태영역을 가지고 있으며, 이는 그 명령어가 수행이 될 것인지 아닌지를 결정하게 된다. 상태영역을 지닌 명령어는 단지 그 상태가 만족되었을 경우에만 수행이 되며, 수행으로 인한 상태 flag의 변경은 일어나지 않는다. 이러한 형태는 파이프라인 프로세서의 주된 페널티인 분기의 사용을 줄일 수 있다. 루프나 조건부 명령어로 변환이 불가능한 피할 수 없는 분기는 파이프라인 동작에 페널티를 야기한다. 지연분기를 사용하면 이러한 페널티도 많이 줄어든다. 각각의 지연 명령어는 하나의 지연슬롯을 가지고 있고 컴파일러에서 사용 가능한 명령어를 그 슬롯에 넣음으로 페널티를 줄일 수 있다.

YS-RDSP는 프로그램과 데이터를 위한 메모리를 내장하고 있다. 이런 메모리들은 X메모리와 Y메모리 두 개의 파트로 나뉘어진다. 각 메모리 파트는 4-KByte의 ROM과 4-KByte의 RAM으로 나뉘어지며 전체 8-KByte ROM과 8-KByte RAM에 된다. 이러한 메모리는 3개의 버스로 제어되는데, 각각 I 버스, X버스, Y버스라 부른다. I버스는 32비트 주소버스와 32비트 데이터버스로 되어있다. 이는 외부 메모리를 제어하는데도 사용된다. RISC ALU와 레지스터파일을 위한 명령어 흐름과 데이터 흐름은 I 버스를 통해 이루어진다. 명령어와 데이터가 동시에 요구되어 질 경우에는 약간의 충돌이 일어날 수 있지만 데이터 요청이 없을 동안 명령어들을 미리 페치하는 프리페치 버퍼를 사용하기에 이러한 충돌을 크게 줄일 수 있다. YS-RDSP는 스칼라 프로세서이고 RISC명령어 길이는 16비트이다. 32word의 프리페치 버퍼는 이 문제를 해결할 수 있다. X버스와 Y

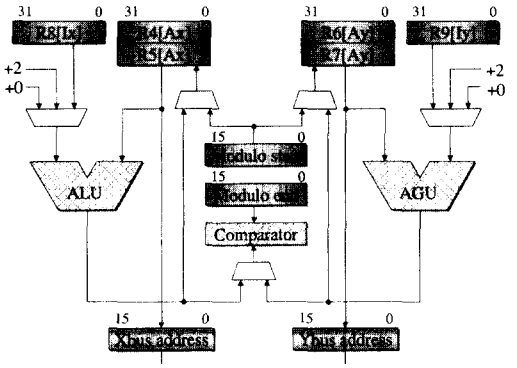


그림 3. 모듈로 주소 생성기

버스는 각각 X메모리와 Y메모리를 제어하기 위한 16-비트 주소버스와 16-비트 데이터버스를 가진다. 그들은 DSP연산 유닛과 DSP레지스터파일에 데이터를 제공하는데 쓰인다. DSP연산은 대개 더 높은 메모리영역을 요구한다. 이 두 개의 데이터버스는 필요한 데이터를 매 사이클마다 전달해 줄 수 있고 이는 높은 성능의 DSP연산을 가능케 한다.

다양한 주소모드는 DSP연산에 필수이다. YS-RDSP에 있는 RISC ALU와 AGU는 두 개의 32-비트 덧셈기로서 뿐만 아니라, 메모리의 병렬 제어를 위한 주소생성기로서의 역할도 수행한다. 자동증가, 자동감소, 참조 모드, 변위모드, PC상대주소모드가 지원된다. 많은 DSP알고리즘에서 유용한 모듈로 주소모드 또한 지원된다. 그림 3은 모듈로 주소모드를 위한 데이터흐름구조를 나타낸다. 모듈로 주소모드가 사용되는 동안에는 메모리를 제어하는 명령어는 해당 주소를 자동증가 시키게 된다. 레지스터 값은 증가시키기 전 주소로 사용되고 이 값을 증가시킨 후 모듈로 종료 주소와 비교된다. 만일 이 주소가 동일한 경우에는 증가된 값 대신 모듈로 시작 주소가 레지스터의 새 값으로 결정된다. 이 모드를 지원하기 위해 모듈로 시작 주소와 모듈로 끝 주소를 저장하기 위한 두 개의 16-비트 레지스터와 추가적인 상태 비트를 요구한다.

RISC 데이터흐름구조는 일반 레지스터 파일에서 값을 읽어 산술, 논리, 쉬프트 동작을 수행하여 결과를 그곳에 저장한다. 일반목적 레지스터 파일은 32-비트 길이의 레지스터 16개로 이루어져 있다. R0에서 R14까지의 레지스터들은 일반 동작을 위해 쓰이고 R15는 스택포인터로의 동작으로 쓰인다. 몇몇 일반 레지스터들은 DSP동작을 위한 주소생성을 위해 쓰이기도 한다.

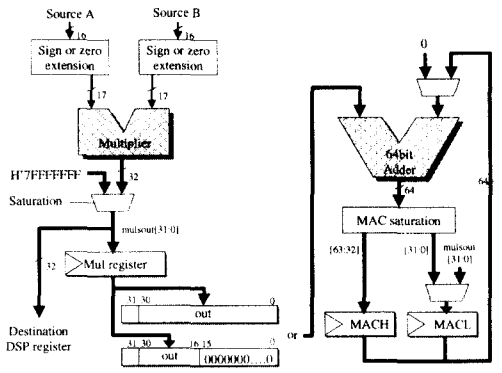


그림 4. MAC유닛의 데이터 흐름도

DSP유닛은 MAC유닛, ALU, 배럴쉬프터로 구성되어 있다. 이는 주로 16비트 고정소수점 데이터 기반으로 동작하며 때로는 32비트와 40비트 데이터도 취급한다. 40비트 데이터흐름을 사용할 경우, 근원 레지스터는 8개의 가드 비트가 직접 사용되며 가드 비트가 없는 경우에는 40비트 길이로 부호확장이 일어난다. MAC유닛 내부의 곱셈기는 부호 없는 곱셈, 부호 있는 곱셈 모두를 지원하기 위해 두 개의 17비트 입력과 33비트 출력을 가진다. 이는 한 사이클에 곱셈을 완료시키기 위해 Radix-4 변형 부스 알고리즘을 사용한다. 그림 4는 MAC유닛의 구체적인 구조를 나타낸다. MAC유닛의 제어신호는 곱셈기로 들어가는 데이터를 선택하고 동작결과를 저장하며 64비트 덧셈기에 결과를 누적하기 위한 입력 데이터를 선택하고 MACH와 MACL의 읽기/쓰기 동작의 제어를 담당한다. 곱셈기는 17비트와 17비트의 곱셈을 수행하며 결과로 34비트를 생성하는데, 부호 비트가 중복되어있기에 실제 곱셈기의 결과는 하위 33비트가 된다. 이는 두 개의 32비트 수를 곱하여 하나의 64비트 값으로 생성시키는 것도 가능한데, 이는 네 번의 17비트 곱셈으로 처리할 수 있다[5]. 이 동작은 4클록 사이클을 소모하며 그 결과는 MACH와 MACL 레지스터에 기록된다. 배럴쉬프터는 다양한 양의 쉬프트동작을 한 사이클 내에 가능케 하기 위해 DSP유닛에 포함되어있다. 산술쉬프트연산의 경우 -32~+32사이의 값만큼을, 논리 쉬프트 연산의 경우 -16~+16 사이의 값만큼을 쉬프트 할 수 있다. 쉬프트 값이 양수일 경우에는 왼쪽 방향을, 음수일 경우에는 오른쪽방향을 나타낸다.

YS-RDSP의 DSP연산은 포화연산모드를 지원한다. 고정 소수점 연산은 오버플로우나 언더플로우로 인한 턴-오버 결과를 야기할 수 있기 때문에 포화연산모드를 사용하여 이러한 문제를 해결해 준다. 오버플로우의 경우 결과 값은 현재 범위에서 표현 가능한 최대의 값을 가지게 된다. 마찬가지로 언더플로우의 경우의 결과 값은 현재 범위에서 표현 가능한 최소의 값을 가지게 된다.

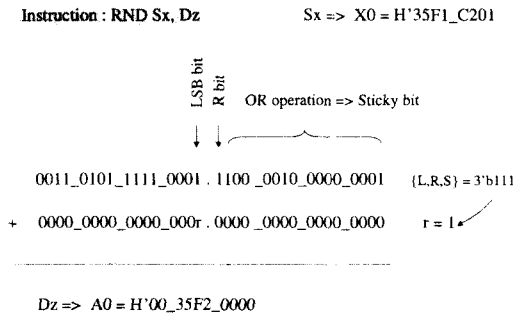


그림 5. YS-RDSP의 라운딩 연산

DSP연산은 32비트 고정 소수점 데이터를 바탕으로 수행되며 단지 상위 16비트만을 메모리에 저장하게 된다. 늘 하위 16비트를 버리게 되므로 많은 경우의 DSP알고리즘에 있어서 동작의 수많은 반복으로 인해 큰 정밀도 손실을 가져오게 된다. 이를 막기 위해 YS-RDSP는 라운딩 명령을 지원하는데, 이 동작은 그림 5와 같다. YS-RDSP에서의 라운딩 방법은 round-to-nearest-even 방식과 비슷하다[6]. 이는 더 높은 정밀도를 적은 하드웨어 부담을 가지고 달성할 수 있게 한다.

YS-RDSP는 칩 내부에 3개의 주변장치를 가지는데, 시스템 제어기와 인터럽트 제어기, 버스 제어기가 바로 그것이다. 동작의 제어를 위해 각각의 주변장치는 메모리 영역에 매핑된 자체 레지스터를 가지고 있고 버스 제어기에 의해 제어되는 주변장치 버스를 통해 접근이 가능하다. 시스템 제어기는 각 유닛에 제공되는 클록 신호를 생성하고 제어레지스터 값에 따른 동작 주파수를 결정한다. 이는 또한 저전력모드를 지원하기 위해 세 종류의 전력강하 모드를 지원한다; SLEEP모드는 CPU코어에 연결된 클록 신호를 중단시키고 STANDBY모드는 CPU와 모든 주변장치의 클록을 중단시키며, MODULE STANDBY모드는 프로그램 수행에 불필요한 몇 개의 주변장치로의 클록 신호를 중단시킨다. 인터럽트 제어기는 외부 디바이스와 주변장치의 인터럽트 요청 우선순위를 결정하고 CPU코어로의 인터럽트 요청을 생성시킨다. 인터럽트들은 16단계의 우선순위를 가지며, NMI(Non-Maskable Interrupt)는 단계 16의 우선순위를 가지게 된다. 버스 제어기는 외부

메모리와의 인터페이스를 관리한다. 이는 SRAM과 DRAM, pseudo-SRAM, burst ROM을 외부 메모리로 지원할 수 있다. 버스 제어기는 내부 제어레지스터와 외부 신호를 통해 프로그램된 외부 접근을 위한 대기 사이클을 제어한다.

Ⅲ. 사이클 단위 시뮬레이터의 작성

검증 및 성능 개선을 목적으로 YS-RDSP와 흡사한 동작을 수행할 수 있도록 C언어를 통해 사이클 단위로 중간 수행 단계를 확인할 수 있는 시뮬레이터가 설계되었다[7].

본 시뮬레이터는 기본적으로 5단계 파이프라인의 각 단계를 구현해 준 부분과 이를 제어하는 부분, 예외 처리 부분, 외부 인터페이스 부분으로 나뉘어지며, 특히 EXECUTE단과 WB_DSP단은 처리해 줄 명령어들을 모아 둔 RISC명령어 모듈과 DSP명령어 모듈이 따로 존재하여 이의 호출을 중심 동작으로 하도록 구현되어 있다. 이 부분은 덧셈기, 곱셈기, 논리 연산 수행부 등으로 구성되어 있다. 외부 인터페이스 부분은 사용자의 편의를 위해 결과를 표시해 주는 부분과 사용자의 제어 입력을 받아들이는 부분으로 구성되어 있는데, 시뮬레이터의 동작을 제어할 수 있는 사용자 인터페이스 모듈, 수행중인 시뮬레이터의 상태를 보여주는 출력 모듈과 외부 입력에 따라서 시뮬레이터의 동작을 제어할 수 있도록 하는 모듈로 구성되어 있으며 추가적으로 외부 파일로 되어 있는 메모리의 상태, 특히 ROM에 기록되어 있는 프로그램을 내부 메모리로 읽어 들일 수 있도록 해 주는 메모리 초기화 모듈 등을 포함하고 있다.

이 중에서, 파이프라인을 제어하는 부분이 이 시

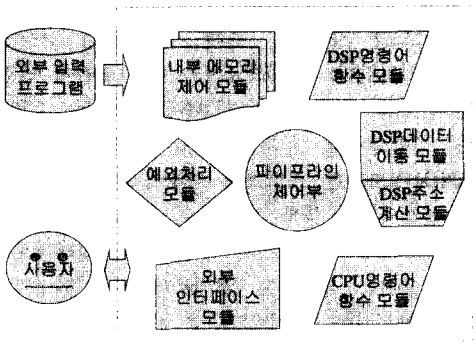


그림 6. 시뮬레이터의 구조도

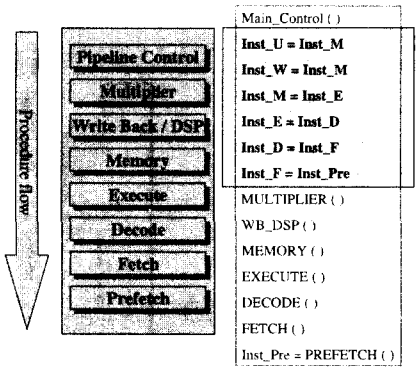


그림 7. 시뮬레이터 상의 파이프라인 흐름 제어

뮬레이터의 중심부라고 할 수 있는데, 이는 각 단계를 호출하는 부분과 각 단계의 수행 중 생성된 제어신호를 처리해 주는 부분으로 구성되어 있다.

파이프라인의 흐름 제어는 우선적으로 명령어의 단계별 수행으로 시작된다. 이는 이전 사이클에서 수행된 명령어를 다음 단계의 명령어 버퍼로 전달하는 방식으로 이루어지게 되는데 그 순서를 원래 파이프라인의 수행단계와는 역순으로 배열하게 된다[8]. 이러한 순서의 제어는 소프트웨어 파이프라이닝과 같은 원리에 의해 실행되는데, 이는 순차적인 실행에 의해 이전 명령어와 충돌이 일어나지 않도록 해 주는 것을 가능케 한다.

그리고, 명령어의 요구에 의한 경우와 메모리 등의 자원의 사용이 중복될 경우에 일어나게 되는 파이프라인의 STALL과 FLUSH에 대한 제어를 수행한다.

우선 STALL제어 방법을 LDC.L (load control register from memory)의 예를 들어 설명해 보면 다음과 같다. LDC.L은 두 사이클의 STALL 후 다음 명령어가 수행되도록 되어있다. 이를 구현하기 위해 DECODE단에 LDC.L명령어가 들어온 경우 FETCH단에 들어갈 명령어를 두 사이클 동안 받아들이지 않고 이전 명령어를 그대로 유지할 수 있도록 해당 단계의 명령어 버퍼를 조작해 주고 DECODE단에 다음 사이클과 그 다음 사이클에 입력되는 명령어는 수행치 못하도록 이 때의 DECODE단의 명령어 버퍼에 NOP을 입력한다. 이의 상태는 그림 8과 같이 표현된다.

다음으로 FLUSH제어 방법을 BT(branch if T is true)의 예를 들어본다. BT는 이 명령어 다음에 FETCH단에서 오는 두 명령어를 버리고 다음에 분기 목적 주소에서 읽어 들인 명령어부터 정상 수행

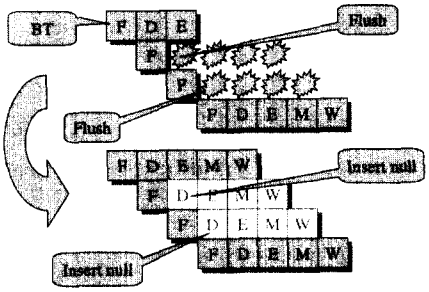


그림 8. Stall 신호 제어의 예

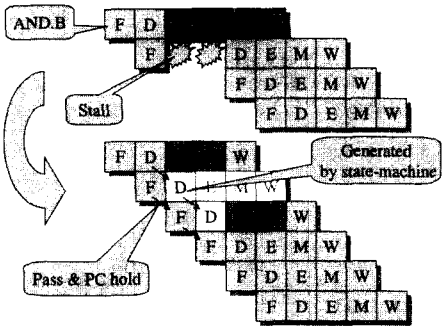


그림 10. 다중 사이클 명령어 제어의 예

하도록 되어 있다. DECODE단에 BT명령어가 들어온 경우 DECODE단에 들어갈 명령어를 두 사이클 동안 수행치 못하도록 DECODE단의 명령어 버퍼에 다음 두 사이클 동안 NOP을 입력하도록 구현한다. 이의 상태는 그림 9와 같이 나타난다.

다중 사이클 명령의 제어 방법의 예로는 그림 10과 같이 AND.B(and byte in memory)를 들 수 있다. AND.B는 이 명령어 다음에 오는 명령을 두 사이클 동안 STALL시키고 다단 명령의 각 단을 수행해 주게 되어 있다. 이를 위해 DECODE단에 AND.B 명령어가 들어온 경우 FETCH단과 DECODE단에 들어온 명령어들을 두 사이클 동안 고정 시켜둔다. 그리고 AND.B를 입력받은 DECODE단은 AND.B를 위한 스테이트머신을 돌린다. 세 사이클동안 AND.B라는 같은 명령어를 받게 되는 EXECUTE단에서는 이를 통해서 각 스테이트마다 마련된 해당 명령어에 대한 다단계 명령어 함수를 호출하여 수행토록 해 준다. 그리고 마지막 스테이트를 수행하게 될 경우에는 EXECUTE단 마지막에서 스테이트머신을 초기화 해 준다.

본 연구에서 설계한 시뮬레이터는 Turbo C++ 3.0상에서 구현하여 DOS환경에서 동작하도록 구현되었고 컴파일 시에 메모리 모델로는 huge을, 연산 시에는 80287 co-processor 명령을 사용토록 하였다. 이의 테스트는 어셈블리 명령어들을 조합해서 만든 프로그램들을 바이너리 파일로 만들어 이를 ROM파일로 만들어 시뮬레이터 상에서 로드하여 사용하도록 하였다.

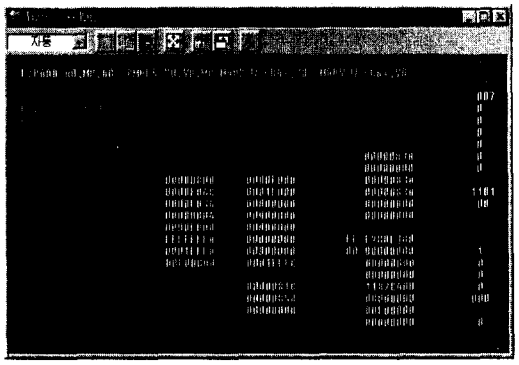


그림 11. FIR 프로그램을 수행시키는 시뮬레이터

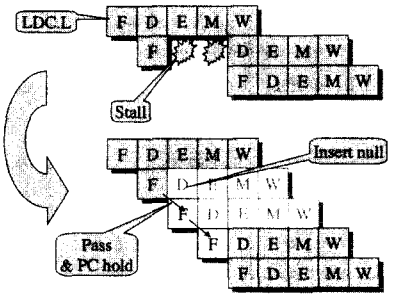


그림 9. Flush 신호 제어의 예

Ⅳ. 성능분석 및 검증

YS-RDSP프로세서는 Verilog-HDL언어로 하향식 설계방식으로 설계되었다. 이의 검증은 먼저 각각의 목표 블록에 대한 입출력신호로 테스트 벡터를 가하여 진행되었다. 또한 C언어를 통해 YS-RDSP와 흡사한 동작을 수행할 수 있도록 사이클 단위로 중간 수행 단계를 확인할 수 있는 시뮬레이터가 설계되었고 전체 프로세서의 동작을 검증하기 위해 이

표 1. YS-RDSP 성능 비교 (단위:사이클 수)

Program	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
FIRnoLoad	5868	13412	3729
JPEGDCT	3821	3787	3086

표 2. YS-RDSP 특성

Process	3.3V, 0.6um, TLM CMOS
Gate count	113291(except memories)
Operation Frequency	41.66 MHz (typical)
Power Dissipation of Core	538 mW
Chip Size	12.1 mm X 9.8 mm
Core Size	10.7 mm X 8.4 mm
Pin number	Total 128 - Input : 9 - Output : 35 - In-out : 32 - Vdd : 26 - Vss : 26

시뮬레이터와의 중간 출력 및 최종 출력 비교가 이루어졌다. 각 명령어에 대한 전체 프로세서의 동작을 검증한 후, 설계된 모델에서 응용 프로그램이 수행되었다. 성능향상을 위해 시뮬레이터의 구조 변경을 통한 테스트가 이루어졌고, 향상된 결과를 보인 구조변경은 HDL모델에 그대로 적용되었다.

성능향상을 위해 쓰인 루프버퍼는 프리페치 버퍼로도 쓰이는데, 이는 기본 성능 뿐 만 아니라 잦은 루프를 갖는 프로그램에 대해 성능향상을 가져오지만, 전체 면적의 증가를 불러온다. 따라서, 여러 정수연산 프로그램과 DSP프로그램을 통한 수회의 시뮬레이션을 거쳐 크기 대 성능비가 가장 좋은 루프버퍼 크기인 32개로 결정하여 적용하였다. 결과적인 성능 분석 또한 방금 언급한 여러 정수연산 프로그램과 DSP프로그램을 통해 이루어졌으며, 표 1은 DSP응용 프로그램들의 성능분석결과를 나타낸다.[9]

설계결과의 검증이 끝난 후, HDL모델은 물리 게이트의 표준-셀 라이브러리를 이용하여 합성되었다. 타이밍분석 및 검증이 출력된 게이트의 네트리스트

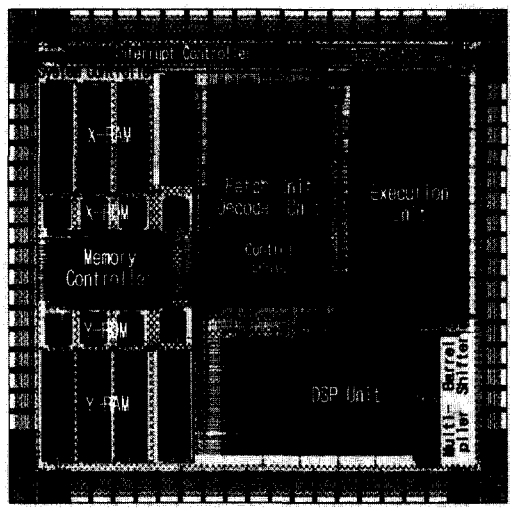


그림 12. YS-RDSP 레이아웃

를 가지고 수행되었다. YS-RDSP는 SYNOPSYS와 COMPASS툴에서 지원하는 3.3V 0.6um CMOS 표준 셀 라이브러리를 통해 합성되었다. 타이밍 분석과 검증은 COMPASSQSIM을 통해 이루어졌다. YS-RDSP는 자동 배치배선을 통해 레이아웃 되었고 결과적인 배치설계도는 그림 12와 같다.

표 3은 YS-RDSP와 요즘 쓰이고 있는 내장형 프로세서들의 사양을 정리한 것이다.[10] YS-RDSP가 내부에 DSP기능을 지원하기 위한 내장 메모리를 사용하였고, 0.6um 설계 공정과 설계방식이 주어진 표준 셀 만 사용했다는 것을 감안하면 칩 크기에서 불리하지 않고 전력소모 면에서도 우수한 성능을 보임을 알 수 있다. 또한 저전력모드를 지원하기에 저전력모드로 들어가면 IBM의 403GCX의 전력소모와 비슷한 값을 보였다.

표 3. 내장형 프로세서의 제품사양

Embedded Processor	Motorola 68EC060	IBM 403GCX	AMD 29040	YS-RDSP
Operating Frequency	50MHz	66MHz	50MHz	42MHz
Power Supply Voltage	3.3V	3.3V	3.3V	3.3V
Power Dissipation	3W	0.4W	1.7W	0.78W
Chip Size	103㎟	n/a	119㎟	118.58㎟
IC Technology	0.42um, 3M	0.45um, 3M	0.7um, 3M	0.6um, 3M

V. 결 론

본 연구에서는 RISC 마이크로프로세서구조와 고정소수점 DSP프로세서구조가 혼합된 YS-RDSP 프로세서 구조를 설계했다. YS-RDSP는 양쪽의 장점을 채용하여 높은 신호처리능력 뿐만 아니라 높은 제어능력과 programmability를 가진다. 함께 개발된 시뮬레이터는 사이클 단위로 동일한 결과가 나오도록 하여 검증을 위한 시뮬레이션 및 구조 개선을 위한 시뮬레이션에 사용되었다. 전체 시뮬레이션 결과는 YS-RDSP가 일반적인 DSP프로세서보다 DSP 응용 프로그램 수행 시 더 적은 사이클이 걸리는 것을 보여주며, 특히 짧은 루프를 갖는 응용에서 더 나은 성능을 가짐을 보여준다. 가변적인 명령어 길이로 인해 YS-RDSP는 32비트 고정길이 명령어를 사용하는 다른 프로세서보다 더 작은 프로그램 코드 크기를 필요로 한다. YS-RDSP는 온칩 시스템 제어기를 통해 낮은 전력 소모를 위한 세 종류의 전압강하 모드를 지원한다. 결과적으로, YS-RDSP는 제어능력, DSP능력과 함께 저전력 소모를 요구하는 이동전화나 PDA 같은 개인휴대단말기에 응용 가능하다.

참 고 문 헌

[1] James Turley, "Selection a High-performance Embedded Microprocessor," *MicroDesign Resources 2nd edition*, pp. 189-218, 1997.

[2] Michael Dolle and Manfred Schlett, "A Cost-Effective RISC/DSP Microprocessor for Embedded Systems", *IEEE Micro*, pp.32~40, Oct. 1995.

[3] Buyer's Guide to DSP Processors, *Berkeley Design Technology Inc.*, 1994.

[4] Phil Lapsley, Jeff Bier and Amit Shoham, DSP Processor Fundamentals, *the Institute of Electrical and Electronics Engineers, Inc.*, 1997.

[5] 이용석, "고성능 마이크로프로세서 곱셈기의 구조", *고성능 마이크로프로세서 구조와 설계 강좌 시리즈*, 1998

[6] IEEE Standard for Binary Floating-Point Arithmetic, *ANSI/IEEE Standard* 754-1985.

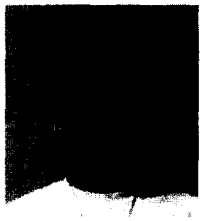
[7] Rozenblit, Co-design: Computer-Aided Software / Hardware Engineering, *IEEE press*, 1995.

[8] David A. Patterson and John L. Hennessy, Computer Architecture: A Quantitative Approach Second Edition, *Morgan Kaufmann Publishers, Inc.*, pp.293~299, 1996.

[9] Alan V. Oppenheim, Ronald W. Schafer, Discrete-time Signal Processing, *Prentice Hall Inc.*, 1989

[10] "Chart Watch : Embedded Processors", *Microprocessor Report*, Vol.12, No.12, SEP.14, 1998.

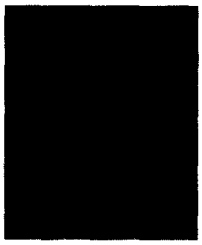
김 문 경(Moon-Gyung Kim) 준회원



1997년 2월 : 연세대학교 전자공학과 졸업
1999년 2월 : 연세대학교 전자공학과 석사
1999년 3월 ~ 현재 : 연세대학교 전기전자공학과 박사과정

<주관심 분야> 마이크로프로세서 설계, SMT, 영상처리

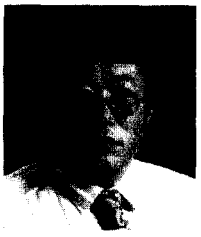
정 우 경(Woo-Kyeong Jeong) 준회원



1996년 2월 : 연세대학교 전기공학과 졸업
1998년 2월 : 연세대학교 전자공학과 석사
1998년 3월 ~ 현재 : 연세대학교 전기전자공학과 박사과정

<주관심 분야> 마이크로프로세서 설계, FPU, 영상처리

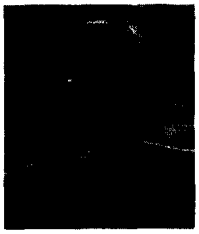
이 용 석(Yong-Surk Lee) 정회원



1973년 2월 : 연세대학교 전기공학과 졸업
1977년 2월 : University of Michigan Ann Arbor 석사
1981년 3월 : University of Michigan Ann Arbor 박사

1993년~현재 : 연세대학교 전기전자공학과 교수
재임 중
<주관심 분야> 마이크로프로세서 설계

이 광 엽(Kwang-Youb Lee) 정회원



1985년 8월 : 서강대학교 전기공학과 졸업
1987년 8월 : 연세대학교 전자공학과 석사
1994년 2월 : 연세대학교 전자공학과 박사
1995년~현재 : 서경대학교 컴퓨터공학과 교수
재임 중
<주관심 분야> 저전력 마이크로프로세서, 암호프로세서