

다중 칩 슈퍼스칼라 마이크로프로세서용 부동소수점 연산기의 설계

이영상*, 강준우**

* 한국과학기술원

305-701 대전광역시 유성구 구성동373-1

** 한국의국어대학교

449-791 경기도 용인시 모현면 왕산리 산89번지

Design of Floating-point Processing Unit for Multi-chip Superscalar Microprocessor

Young-sang Lee* and Jun-Woo Kang**

* Korea Advanced Institute of Science and Technology

305-701 Gusung-Dong, Yusong-Gu, Daejeon, KOREA

** Hankuk University of Foreign Studies

89 Wangsan, Mohyun, Yongin, Kyonggi-Do, 449-791, KOREA

Abstract

We describe a design of a simple but efficient floating-point processing architecture exploiting concurrent execution of scalar instructions for high performance in general-purpose microprocessors. This architecture employs 3 stage pipeline asynchronously working with integer processing unit to regulate instruction flows between two arithmetic units.

본 논문에서는 64 비트 연산 경로와 파이프라인 구조를 갖는 다중 칩 슈퍼스칼라 마이크로프로세서에 사용될 고성능 부동소수점 연산기의 구조를 제안하였다. 이 연산기는 SPARC-V9 [2]의 부동소수점 연산 명령어를 처리하는 3 단계 파이프라인 연산기로서 범용 연산기에 내장되어 상호 비동기화를 이루어, 슈퍼스칼라 방식의 정수처리기와 자연스럽게 동작하도록 설계되었다. 또한, 데이터 일관성을 위하여 정수 연산기에 리오더 버퍼를 사용함으로써 파이프라인의 제어를 간편하게 하였으며 레지스터 파일과 같은 대량의 하드웨어를 정수 연산기로 옮겨 하드웨어 가격을 줄였다.

1. 서론

부동소수점 연산기는 다차원 그래픽, 음성, 영상을 포함한 멀티미디어 데이터의 처리 및 공학적 시뮬레이션에 많이 사용되고 있으며, 병렬화된 연산 처리부와 확장된 데이터 경로를 사용하는 대부분의 고성능 마이크로프로세서에 내장되어 사용되고 있다.

2. 마이크로프로세서의 구조

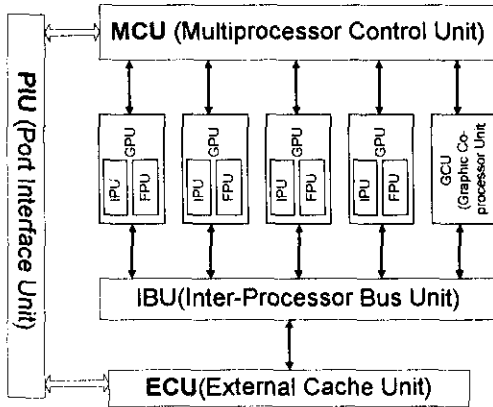
Raptor 라고 불리는 다중 칩 슈퍼스칼라 마이크로프로세서는 4 개의 독립된 범용 연산기 (GPU : General Processor Unit)외에 그래픽 연산기(GCU), 제어기, 접속부, 버스 및 메모리부로 이루어져 있으며 각각의 범용 연산기는 서로 다른 쓰레드를 수행한다. [3]

반도체 공정 기술의 발달로 집적도가 꾸준히 증대되고 있음에도 불구하고 각종 연산기와 데이터의 효과적인 제어를 위해서는 슈퍼스칼라 [1]와 같은 복잡한 제어 방법이 사용되며, 마이크로프로세서에 내장되는 각종 소자의 종류와 기능이 증가함에 따라 단일 마이크로프로세서 칩에 내장되는 부동소수점 연산기는 단순한 제어 구조와 최소 면적을 차지하면서도 고성능을 갖도록 설계되어야 한다. 또한, 슈퍼스칼라나 다중 쓰레드를 지원하는 소프트웨어와 하드웨어의 등장으로 이에 적합한 구조를 갖는 부동소수점 연산기가 필요하게 되었다.

각 범용 연산기에는 정수 연산기(IPU), 부동소수점 연산기(FPU), 디코더, 레지스터 파일등의 주요 연산단위가 있으며 그외에 리오더 버퍼, 레저베이션 스테이션(Reservation Station), 명령어 캐쉬, 결과 버스 중재 유닛(Result Bus Arbitration Unit)등이 있다. 그래픽 연산기(GCP)는 4 개의 범용 연산기에 의하여 공유되며

IBU(Inter-processor Bus Unit)에 연결되어 있다.

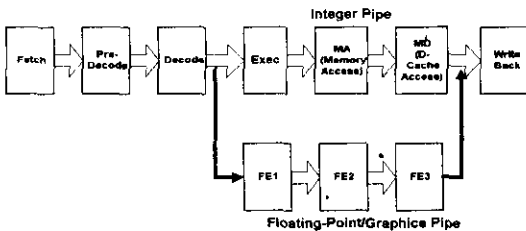
범용 연산기는 비순차적 완료 (Out of Order Completion)을 지원한다. 부동소수점 연산기를 포함한 다중 슈퍼스칼라 마이크로프로세서의 구조는 다음 (그림 1)과 같다.



(그림 1) Raptor 마이크로프로세서의 구성

3. 부동소수점 연산기의 연산 구조

부동소수점 연산기가 처리할 부동소수점 데이터의 형식은 IEEE-754 표준 [4]의 32 비트와 64 비트 데이터 형식에 따른다. 구현될 명령어는 공개 구조를 갖는 SPARC V9 구조 중 부동소수점 명령어들이다. [2] 부동소수점 연산기의 파이프라인은 (그림 2)와 같다. 정수 연산기의 파이프라인은 연산명령어 실행시와 로드/스토어 유닛의 작동에 따라 단계가 가변적이므로, 부동소수점 연산기의 파이프라인은 정수 연산기와는 비대칭적으로 운용되며 핸드 셰이킹 방법을 이용한 비동기적인 방식으로 상호접속을 이루고 있다.



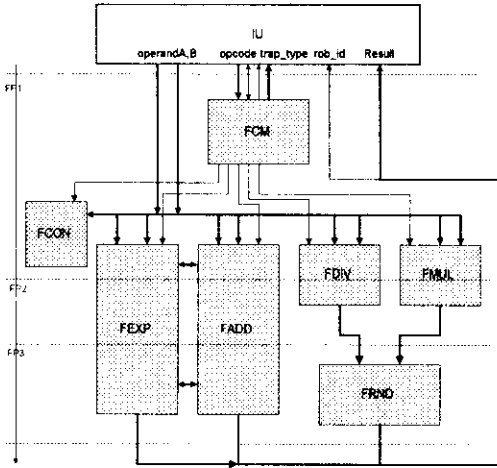
(그림 2) 연산기의 파이프라인 구조

부동소수점 연산기의 파이프라인의 1 단계에서는 부분 디코드된 연산자와 피연산자, 그리고 리오더 버퍼에 관한 정보를 받는다. 즉, 이 단계에서 부동소수점 연산기 내부의 제어부로 전달된 명령어는 완전 해석을 통하여 적절한 제어 신호를 발생한다. 이것은 전체 사이클 중 앞 부분에서 일어나야 한다. 왜냐 하면, 사이클의 나머지는 3 단계 부동소수점 연산의 첫단계를 수행해야 하기 때문이다. 이 단계에 해당하는 것은 전처리(pre-processing)과정으로서 부동소수점 형식으로 표현된 데이터를 부동소수점 연산기에 입력하기 위해 필요한 피연산자의 지수 부분과 가수 부분을 분리해서 각 연산 모듈에 입력하며, 곱셈의 경우 radix 의 부호화와 곱셈 트리를 만든다. 제 2 단계에서는 실제 연산이 행해지며 부분적인 결과가 생성된다. 즉, 곱셈의 경우는 부분 곱(partial product)을 모두 더하는 52 비트 CSA 덧셈기가 덧셈을 수행한다. 제 3 단계에서는 연산의 후처리(post-processing) 과정으로 라운딩(rounding)을 수행한다. 부동 소수점 나눗셈이나 제곱근 계산의 경우는 전체적인 부동 소수점 파이프라인과 독립적으로 이루어진다.

4. 부동소수점 연산기의 기능 모듈

설계된 부동소수점 연산기는 100MHz 의 단일 위상의 클럭을 기본 동작 주파수로 사용하여 64 비트 덧셈과 곱셈의 경우 1 사이클, 나눗셈의 경우 기본적으로 27사이클(배정도의 경우) 내에 연산을 수행할 수 있도록 설계 되었다. [5]

부동소수점 연산기의 내부 구조를 (그림 3)과 같이 제어부, 연산부의 2부분으로 나누어 지며, 특히, 이 부동소수점 연산기는 레지스터 파일을 내부에 갖지 않는다. 이것은 정수연산기가 다중으로 동작하는 슈퍼스칼라 구조에서 레지스터 파일을 정수부와 부동소수점 연산부를 공통으로 관리함으로써 예외 상황의 발생시 리오더 버퍼를 이용하여 원래 레지스터의 상태로 복귀시키는 작업을 간단하게 수행하기 위한 것이다. 레지스터 파일을 사용하지 않는 대신 부동소수점 연산 결과마다 태그에 해당하는 리오더 버퍼 인식자를 정수 처리기에 보냄으로써 각 연산을 구분한다.



(그림 3) 부동소수점연산기의 구조

상태 레지스터(Floating-point Status Register, FSR) 또한 정수연산기가 관리를 한다. 다만 이의 관리에 필요한 정보는 정수연산기와 부동소수점 연산기간의 데이터 교환을 통해 전달된다. 예를 들어, 예외 상황의 발생 시 부동소수점 연산기는 예외상황을 일으킨 원인을 8 비트의 데이터라인을 이용하여 전송하고 이를 정수연산기가 상태 레지스터에 저장한다.

부동소수점 제어 모듈(FCM)은 정수 연산기와의 비동기 접속을 위한 제어신호, 부분 해석된 연산자 코드, 결과값을 쓰기 위한 구분자(ROB identifier) 등을 받아서 부동소수점 연산 명령어들이 들어온 순서대로 해당하는 각 연산 모듈로의 제어 신호를 발생 시키며, 부동소수점 연산 명령어들의 수행 파이프라인을 제어 하는 신호를 발생시킨다. 제어 모듈은 제산/제곱근 연산 모듈이 연산을 수행 중일 경우 더 이상의 명령어 처리를 거부한다. 따라서, 파이프라인은 이미 들어온 연산을 다치면 제산/제곱근 연산이 끝나서 후처리부로 넘어올 때까지 정지된 상태가 된다. 이는 정수연산부의 리오더 버퍼의 크기와 제어의 효율성을 고려한 것이다.

부동 소수점 가산 모듈(FADD)은 부동소수점 피연산자를 정렬시키고 피연산자의 보수 및 상수간의 덧셈과 뺄셈을 수행하며 마지막 단계에서는 라운딩 작업을 한다. 부동소수점 지수 계산 모듈(FEXP)은 피연산자의 지수들을 비교하고 덧셈, 뺄셈, 보수화, 쉬프트할 비트 수의 계산 등을 수행한다. 부동소수점 승산

모듈(FMUL)은 피연산자의 상수의 곱셈을 수행하는데, 연산 결과값의 정규화(normalization) 수행과 아울러 부동소수점 라운드 모듈(FRND)과 함께 연산 결과값을 라운드시킨다. 부동소수점 라운드 모듈은 곱셈 뿐만 아니라 제산/제곱근 제산 결과 값의 라운딩도 수행하며, IEEE 754 표준에 정의된 라운딩 모드를 지원한다.

부동소수점 제산/제곱근 연산 모듈(FDIV/SQRT)은 부동소수점 나눗셈과 제곱근 계산을 동일한 하드웨어를 사용하여 수행한다. 부동소수점 제어부는 하드웨어(Hardwired)방식으로 설계되었다.

제산/제곱근 연산은 다른 연산의 파이프라인과는 별도로 동작되며 파이프라인의 마지막 단계에서 전후로 실행된 명령과 동기화 된다.

5. 정수연산기와의 접속 방식

정수 연산기와 부동소수점 연산기 간의 접속(interface)은 hand shaking 방식으로 이루어진다. 부동소수점 연산기는 리오더 버퍼를 갖는 정수 연산기와 연산자, 피연산자, 연산 결과값 및 제어 신호를 교환할 때 리오더 버퍼의 용량과 데이터의 공급/처리 속도등을 맞출 수 있도록 제어 신호를 교환한다. 이러한 제어 신호는 3 가지로 분류된다. 첫째는 입력 제어 신호로서, 4 개의 정수 연산기에 할당된 부동소수점 연산기는 dispatch 신호를 정수 연산기로부터 받는 것으로 연산 코드와 데이터를 받아들인다. 이 때, 부동소수점 연산기가 연산이 종료된 결과값을 갖고 있으나 쓰지 못하고 있거나 예외 상황의 발생으로 더 이상의 진행이 무의미할 때 등을 제외하고는 ready 신호를 발생하여 데이터를 넘겨 받는데 아무 문제가 없음을 정수연산기에 알린다. 둘째는 출력 제어 신호가 있다. 여기에는 부동소수점 연산기에서 연산이 종료된 경우, 그 결과값을 정수 연산기에 넘길 때를 알려주는 wreq(Write Request)신호와 정수연산기가 결과값을 받을 수 있는 상황인지 아닌지를 알리는 wgnt(Write Grant)신호가 있다.

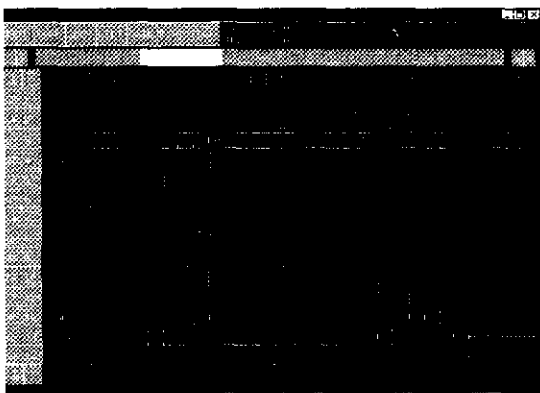
셋째는 예외 상황 제어 신호이다. 예외 상황이 발생되면 부동소수점 연산기는 trap 신호를 발생시키고 동시에 예외상황의 종류를 8 비트의 데이터경로를 통하여 정수 연산기에 알린다. 이 경우, 정수 연산기로부터의 확인(Ack) 신호는 없다. 다음은 부동소수점 연산기와 정수 연산기의 접속신호를 정리한 표이다.

<표 2> 부동소수점 연산기의 접속 신호

신호명	I/O	설 명
Clk, reset	I	클럭 및 리셋
rob_id	I	리오더 버퍼 인식자(tag)
rob_id_rslt	O	리오더 버퍼 인식자 출력값
ftt[8]	O	예외 상황의 종류
fpu_trap	O	외부에서 예외 상황이 감지 되었음을 정수 처리부에 보냄
dispatch	I	명령어 보내기를 나타낸다
ready	O	연산기의 가동 여부 알림
opcode[8]	I	부분 해석된 명령어
operandA operandB	I	피연산자 A,B
result	O	연산 결과값
wgnt	I	결과치를 써도 좋음
wreq	O	결과치를 써도 좋은지 허가 신호

발생된 예외 사항들은 파이프라인의 예외 상황 처리 단계에서 한번에 1 개씩 처리된다. 부동소수점 연산기는 5 가지의 IEEE-754 의 라운딩 모드를 지원하는데 이에 대한 선택은 상태레지스터에 지정된 라운딩 비트에 따라서 수행된다. [7]

설계된 부동소수점 연산기는 Verilog 로 모델링되어 명령어들의 수행이 시뮬레이션 되었다. 각 신호에 대한 타이밍 결과는 다음 (그림 4)와 같다.



(그림 4) 타이밍 시뮬레이션 결과

6. 결론 및 향후 계획

본 논문에서 기술한 부동소수점 연산기는 슈퍼스칼라를 지원하는 다중 프로세서 칩을 구성하는 4 개의 범용 연산기에 내장되어 슈퍼스칼라 구조 아래에서 간단한 제어를 실현하기 위한 구조를 가지면서도 효율적인 데이터의 연산을 제공한다. 부동소수점 연산기는 정수 연산기와 핸드 웨이킹 방식의 비동기 접속을 통하여 데이터와 제어 신호를 주고 받음으로써 상호 간의 데이터 흐름을 효과적으로 조절한다. 또한, 연산 단위끼리의 하드웨어 공유를 통하여 다중 칩의 효율성을 높이고 있다. 데이터 일관성 (integrity)를 위한 모뎀 제어 기능을 정수 연산기에서 운영하며, 레지스터 파일을 부동소수점 연산기 외부에 설계하므로써 더욱 작은 부동소수점 연산기의 설계가 가능해 졌다. 설계된 부동소수점 연산기는 현재 behavioral 수준에서 Verilog [9] 로 모델링되어 기능 검증이 완료되었다.

참고 문헌

- [1] M. Johnson, "Superscalar Microprocessor Design," Prentice-Hall, 1991.
- [2] D.L. Weaver and T. Germond, "The SPARC Architecture Manual", Version9, SPARC International Inc.
- [3] 강준우외 2 인, "다중프로세서칩 개발에 관한 연구," 대한전자공학회 추계종합학술대회 논문집, 제 17 권, 제 2 호, pp. 810-813, 1994 년 11 월.
- [4] IEEE Standard for Binary Floating-Point Arithmetic, ANSI/IEEE Std 754-1985.
- [5] J. A. Prabhu and G. Zyner, "167MHz Radix-8 Divide and Square Root Using Overlapped Radix-2 Stages," IEEE Symposium on Computer Arithmetics, p.156, 1995.
- [6] Ted Williams, N.Patkar, G.Shen, "SPARC64: A 64-b 64-Active-Instruction Out-of-Order-Execution MCM Processor", IEEE JSSC Vol.30, No 11.
- [7] UltraSPARC-1 User's Manual, SPARC Technology Business, Revision 1.0, 1995.
- [8] D. J. Kuck, D. S. Parker and A. H. Sameh "Analysis of Rounding Methods in Floating Point Arithmetic," IEEE Transaction on Computers Vol. C-26 No.7, p.643, 1977.
- [9] Verilog-XL Reference Manual, Cadence Design Systems, Version 1.6, 1991.