

부동 소수점 가산기 모듈의 설계와 PCI 인터페이스를 통한 검증

정명수, 손승일

한신대학교 정보통신학과

Design of Floating Point Adder and Verification through PCI Interface

Myung Su Jung, Seung Il Sonh

Dept. of Information and Communication HanShin University

E-mail : redpunkki@hotmail.com

요 약

수치연산 보조프로세서로도 알려져 있는 부동 소수점 연산장치(FPU)는 컴퓨터가 사용하는 기본 마이크로프로세서보다 더 빠르게 숫자를 다룰 수 있는 특별한 회로 설계 또는 마이크로프로세서를 말한다. FPU는 전적으로 대형 수학적 연산에만 초점을 맞춘 특별한 명령 셋을 가지고 있어서 그렇게 빠르게 계산을 수행할 수 있는 것이다. FPU는 오늘날의 거의 모든 PC에 장착되고 있지만, 실은 그것은 그래픽 이미지 처리나 표현 등과 같은 특별한 일을 수행할 때에 필요하다. 초창기 컴퓨터 회사들은 각기 다른 연산방식을 사용했다. 이에 따라 연산결과가 컴퓨터마다 다른 문제점을 해결하기 위해 IEEE에서는 부동 소수점에 대한 표준안을 제안하였다. 이 표준안은 IEEE Standard 754 이며, 오늘날 인텔 CPU 기반의 PC, 매킨토시 및 대부분의 유닉스 플랫폼에서 컴퓨터 상의 실수를 표현하기 위해 사용하는 가장 일반적인 표현 방식으로 발전하였다. 본 논문에서는 부동 소수점 표준안 중 32-bit 단일 정밀도 부동 소수점 가산기를 VHDL로 구현하여 FPGA칩으로 다운하고 PCI 인터페이스를 통해 Visual C++로 데이터의 입출력을 검증하였다.

1. 서론

우주과학, 그래픽 이미지 처리, 통계 등 오늘날의 복잡한 연산은 모두 컴퓨터가 처리를 한다. 이러한 연산에서는 분수(fraction)와 지수(exponent)를 사용하는 부동 소수점 표현 방식을 널리 사용되고 있다. 수치연산 보조프로세서로도 알려져 있는 부동 소수점 연산장치는 기본 마이크로프로세서보다 더 빠르게 숫자를 다룰 수 있는 특별한 마이크로프로세서를 말한다. 초기의 부동 소수점 수의 표현 형식은 각 기업이나 연구소 등에서 별도로 정의하여 사용하였으나, 1985년에 IEEE 학회에서는 이를 통합하여 표준안으로 마련하였고, 현재 전 세계의 대부분의 기업 및 연구소에서 이 표준안을 따르고 있다[1].

부동 소수점 연산기(Floatint-Point Unit)는 앞에서 말한 IEEE Standard 754 부동 소수점 표준안에 맞게 부동 소수점에 관한 모든 연산을 하는 장치이다. 대형 수학적 연산에만 초점을 맞춘 특별한 명령 셋을 가지고 있어서 빠르게 계산을 수행할 수 있다. 반도체 분야의 발전으로 칩(Chip)의 집적도가 증가함에 따라서 중앙처리 장치

(Central Processing Unit)와 함께 한 칩에 내장할 수 있게 됨으로써, 주 연산기의 중요한 요소로 등장하고 있다.[2].

본 논문에서는 부동 소수점 표준안 중 32-bit 단일 정밀도 부동 소수점 가산기를 VHDL로 구현하여 FPGA칩으로 다운하고 PCI 인터페이스를 통해 Visual C++로 데이터의 입출력을 검증하였다.

II. 부동소수점 표현방법

2.1 단일정밀도

	1	8	23
S	지수(E) 필드		가수(M) 필드

[그림1] IEEE 표준 단일정밀도 형식

그림1의 단일정밀도는 1비트의 부호 비트와 8비트의 지수 비트 그리고 23비트 유효자릿수로 표현된다. 유효자릿수는 소수점의 왼쪽에 1이 있

는 것으로 하며, 정규화 된 유효자릿수는 1에서 2 사이의 값을 갖는다. 지수는 127의 바이어스 사용하는데 그 범위는 '-126 ~ 127'이며, 지수값 중 0000 0000 (-127)과 1111 1111 (128) 은 예약되어 있다.

$$N = (-1)^S 2^{E-127} (1.M) \quad (\text{식 1})$$

부동 소수점 N은 식 1과 같이 나타낼 수 있으며 지수는 $1.M \times 2^E$ 의 형태를 가지며, 소수점 아래 M 부분만 가수 필드에 저장된다. 여기서 소수점 왼쪽의 표현되지 않는 1을 hidden bit라고 지칭한다[1][3][4].

표1은 단일정밀도 형식을 사용하여 부동소수점을 부호, 지수, 가수로 표현하는 예를 보여주고 있다

표1. 단일정밀도 형식 표현

부호(S) 비트 = 1 (음수)
지수 .E = 00000011 + 01111111 = 10000010 (바이어스 127을 더한다.)
가수 M = 101101000000000000000000 (소수점 좌측의 1은 제외한다.)

부동 소수점 N을 -13.625라고 하였을 때, 식 1과 같이 표현하면 식 2과 같이 나타낼 수 있다.

$$13.625_{10} = 1101.1012 = 1.101101 \times 2^3 \quad (\text{식 2})$$

식 3을 이용하여 단일정밀도 형식에 맞게 표현하면 표1과 같다.

표1의 내용을 그림으로 나타낸 것이 그림2이다.

S	E	M
0	10000010	101101000000000000000000

[그림2] 부동 소수점 '-13.625'를 IEEE 표준 단일정밀도 형식으로 표현

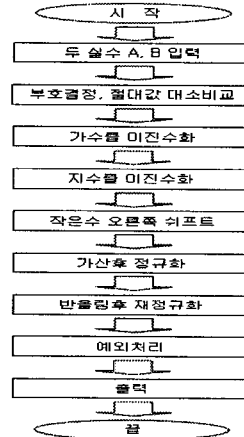
2.2 반올림(Rounding)

2.2.1 Round to nearest Even

가장 가까운 쪽으로 반올림 혹은 반내림하는 방법으로 양쪽이 똑같이 가까운 경우 LSB(Least Significant Bit)가 0이 되도록 하는 방법이다. 이 방법이 가장 많이 사용되는 방법으로 이렇게 하기 위해서는 배럴 쉬프트(barrel shifter)에서 guard, round, sticky 비트를 검출해야 한다[3][5].

III FPA 설계

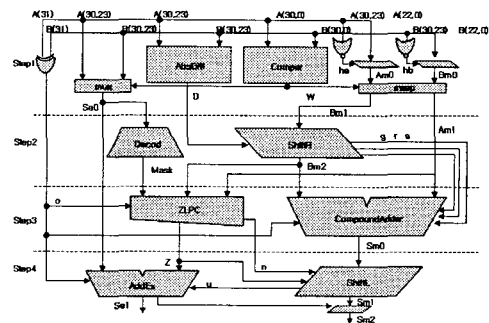
3.1 부동 소수점 가산기의 연산 과정



[그림3] 부동 소수점 가산기의 연산 과정

프로그램의 대략적인 수행과정은 그림3과 같이 우선 부호와 가수, 지수 순서로 분리된 두 실수 A, B를 입력받아서 지수를 통일해주기 위해 작은수를 오른쪽 쉬프트하여 가산을 한다. 그리고 연산결과를 부동소수점 형식대로 정규화를 시킨다. 만일 반올림을 해줄 경우에는 반올림을 하고 재정규화를 실시한다. 연산결과에 따라 예외처리를 하고 출력한다.

3.2 부동 소수점 가산기의 설계 구성



[그림4] 부동 소수점 가산기의 전체 블록도

그림4에서 볼 수 있듯이 부동 소수점 연산과정이 4 step으로 구성되어져 있다.

A와 B 각각의 수로부터 32비트를 입력 받으면 step 1에서 AbsDiff 모듈로부터 지수의 차가 연산되고 Compar모듈에서 부호를 제외한 절대값의 대소비교를 통해 큰 수를 A로 작은 수를 B로

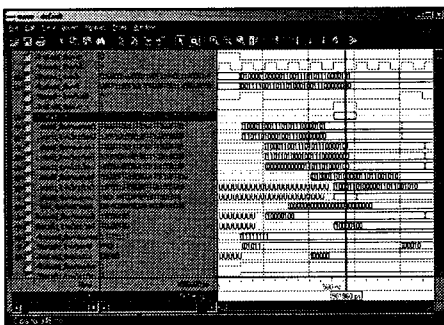
swap한다.

step 2에서는 ShiftR모듈로 절대값이 작은 수를 받아 지수의 차만큼을 오른쪽 쉬프트하고 Guard/Round/Sticky bit를 가지게 된다. Decod 모듈은 두 수를 절대값의 대소비교에 따라 지수 값을 마스크를 통하여 설정해준다.

step 3에서는 ZLPC모듈을 통해 앞에 0의 개수가 몇 개인지 정규화가 되었는지에 대한 정보를 알려준다. CompoundAdder모듈은 부호비트와 가수비트 그리고 나머지 비트들을 가지고 알고리즘에 의해서 가산과 반올림 연산을 하게 된다.

마지막으로 step 4에서는 ShiftL모듈에서 재정규화를 시켜주고 AddEx모듈에서 결과에 따라 지수를 가산시켜준다.

3.3. ModelSim을 이용한 검증

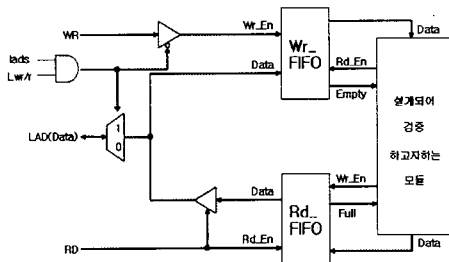


[그림5] ModelSim 출력과형

그림5는 VHDL언어를 사용하여 얻은 부동소수점 가산기의 출력과형을 보여주고 있다. valid 신호가 활성화 되면 두개의 32비트 실수를 입력받아 4 step의 클럭을 거쳐 최종 32비트의 연산결과가 출력된다.

IV. PCI 인터페이스

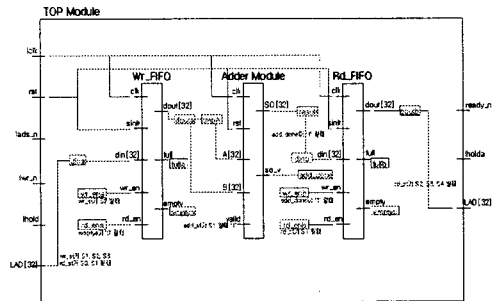
4.1 인터페이스 블록도



[그림6] 인터페이스 블록도

그림6은 Visual C++ 6.0으로부터 부호, 지수, 가수의 32비트로 분리된 부동소수점 형식의 두 실수 A, B를 PCI 인터페이스로 보냈을 때 타겟보드에서 PCI 인터페이스로부터 들어오는 신호의 입출력 흐름을 나타내는 블록도이다.

4.2 인터페이스 세부블록도



[그림7] 인터페이스 세부 블록도

그림7은 그림6 블록도의 신호들을 세부적으로 나타낸 것이다.

TOP모듈 안에는 우선 데이터의 쓰기 버퍼용 FIFO와 읽기 버퍼용 FIFO 그리고 부동 소수점 가산기 모듈이 있다.

PCI 인터페이스를 통해서 두 실수 A, B의 32비트 데이터가 LAD신호를 통해 들어오게 된다.

들어온 데이터는 상태머신과 제어신호에 따라 단계별로 FIFO에 저장되어 있다가 부동 소수점 가산기 모듈에 넘겨주게 되고 연산결과는 다시 FIFO로 저장되어 결국 LAD신호를 통해 다시 PCI 인터페이스를 통해 32비트의 값을 넘겨주게 되는 것이다.

V. 부동소수점 가산기 실행

그림8은 두개의 실수 A, B를 32비트 단일 정밀도의 부동소수점 형식으로 가산하는 결과를 보여주고 있다.

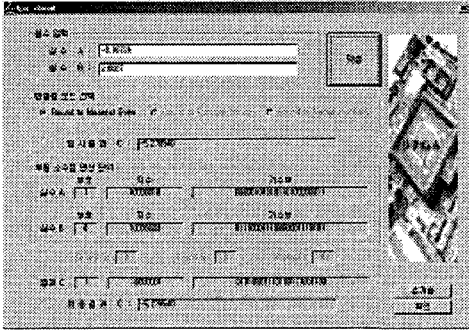
반올림 모드로 Round to Nearest Even방식을 사용하였다.

부호에 상관없이 실수 A와 B를 입력하고 덧셈 버튼을 클릭하면 두 실수가 Visual C++ 6.0으로 연산되어 부동소수점 형식의 부호(1 bit)와 지수(8 bit), 가수(32 bit)로 분류하여 비트열로써 저장한다.

32비트의 두 수는 PCI 인터페이스를 통해 FPGA칩이 있는 타겟보드로 송신되어 FIFO에 저

장되어 있다가 가산을 하게 된다. 가산된 연산결과는 다시 FIFO로 저장되어 있다가 PCI 인터페이스를 통해 32비트의 값을 넘겨준다.

32비트의 부호, 지수, 가수로 된 부동소수점 형식의 가산결과는 다시 실수형식으로 재구성하여 최종 결과를 출력하게 된다.



[그림8] 프로그램 실행 화면

VI. 결론

부동 소수점 가산기는 FPU 설계의 핵심 유닛으로 사용되며 데이터의 수치연산을 하는 일에 다양하게 사용되어 진다. 연산량이 큰 데이터 처리에 주로 이용되기 때문에 불필요한 처리를 되도록 줄여서 설계해야 한다.

본 논문에서는 부동소수점 연산처리에 중요한 요소인 저전력과 고속 처리를 위해 step마다 제어 신호를 두고 현재 step이 아니면 연산하지 않도록 하고 있다. 먼저 부동소수점 형식에 대해 연구하고 연산과정대로 Visual C++ 6.0으로 시뮬레이션을 해서 최종적으로 핵심 연산과정을 VHDL언어를 이용하여 하드웨어로 구현하였다. 그리고 인터페이스를 통한 프로젝트 환경 구축으로 타겟 보드의 송수신을 검증하였다.

설계된 부동소수점 가산기 모듈은 FPU의 다른 모듈들과 함께 단일 코어로 최적화되어서 칩으로 설계되어 DSP분야와 그래픽 프로세싱 분야는 물론 여러 과학 연산 분야 등에서 활용될 수 있다.

참고문헌

- [1] ANSI/IEEE Standard 754-1985 for binary Floating-Point Arithmetic, IEEE Computer Society Press, Los Alamitos, Calif., 1985.
- [2] 박우찬, 정철호, 양진기, 한탁돈, "IEEE 반올림과 덧셈을 동시에 수행하는 부동 소수점 곱셈연산기 설계", 전자공학회논문지, vol.34-C, no.11, 1997.11.
- [3] 이용석, "60MHz Clock 주파수의 IEEE 표준 Floating Point ALU", 전자공학회논문지,

vol.28-A, no.11, 1991.11

- [4] Loucas Louca, Todd A. Cook, William H.Johnson "Implementation of IEEE Single Precision Floating Point Addition and Multiplication on FPGAs", Rutgers University, 1996.9
- [5] 박우찬, 이시화, 권오영, 김신동, 한탁돈, "Floating Point Adder / Subtractor Performing IEEE Rounding and Addition / Subtraction in Parallel", IEICE TRANS.INF. & SYST., vol.E79-D, no.4, April 1996