

SMT 마이크로프로세서에 적합한 ALU의 설계

김 상 철, 홍 인 표, 이 용 석
연세대학교 전기 전자공학과

전화 : 02-2123-2872 / 핸드폰 : 016-564-2625

Design of an ALU for SMT Microprocessors

Sang Cheol Kim, In Pyo Hong, Yong Surk Lee
Dept. of Electrical & Electronic Eng., Yonsei University
E-mail : schkim@dubiki.yonsei.ac.kr

Abstract

In this paper, an ALU for Simultaneous Multi-Threading (SMT) microprocessors is designed. The SMT architecture improves notably performance and utilization of processes compared with conventional superscalar architectures by executing instructions from multiple threads at the same time. This ALU adopts data bypassing method to process multi-threads. And it can flush instructions in the same thread that generate exceptions such as branch misprediction, interrupt etc, performance of SMT microprocessors with data bypassing and exception handler can be improved.

I. 서론

컴퓨터의 핵심 부품인 마이크로프로세서는 집적기술의 발달과 새로운 여러 가지 구조를 도입하면서 더욱 빠른 처리 속도를 가지게 되었다[1]. 고성능의 마이크로프로세서들은 성능의 향상을 위해 ILP(Instruction Level Parallelism)를 이용한 슈퍼스칼라 구조를 채택하고 있는데, 이는 그 자체에 한계점을 지니고 있다. 이러한 한계점을 뛰어 넘을 수 있는 TLP(Thread Level Parallelism)를 이용하기 위한 여러 가지 방안들이 제시되어 왔다. 그 중에 하나로 Simultaneous Multi-Threading(SMT) 구조가 제안되었는데, 이러한 SMT 구조는 ILP 와 TLP 의 특성을 최대한으로 활용

함으로써 그 성능의 향상을 꾀하고 있다[2].

본 논문에서는 본 연구실에서 연구된 순차적 SMT 마이크로프로세서 구조를 바탕으로 최적화된 ALU를 설계한다. 설계된 ALU는 SMT의 특징을 구현하기 위해 새롭게 파이프라인 되고, 다중스레딩의 구현을 위한 각각의 독립적인 스레드를 처리하기 위해 스레드 개수만큼의 레지스터 파일과 스코어보드 어레이와 데이터 인터페이싱을 하는 특징을 갖는다. 특히 스코어보드 어레이와 ALU는 데이터 바이패싱을 가능하게 하는 특징을 갖는다. 분기 예측 오류의 페널티(branch misprediction penalty)를 줄이기 위하여 명령어는 조건부 실행의 특징을 갖고 있다. 그리고 다중 스레딩 환경에서 예외 처리, 분기 실행 등이 가능하다.

II. SMT 마이크로프로세서

오늘날, 고성능 마이크로프로세서들이 대부분 성능 향상을 목표로 채택하고 있는 구조는 슈퍼스칼라구조이다. 이것은 ILP를 이용한 것이지만, 그 자체로 한계점을 가지는데 명령어 이슈 대역폭을 넓힌다 해도 IPC(Instruction Per Cycle)는 2를 넘기기 힘들고, 대역폭을 보다 넓혀도 그 성능의 향상정도는 미미한 수준이라는 것이다[2]. 이러한 시스템의 성능 향상을 위해서, TLP를 이용하기 위한 여러 가지 방안들이 제시되고 있다. 멀티프로세서의 경우가 그 대표적이라 할 수 있으나, 이 경우 자원의 공유가 힘들기 때문에 스레드가 한 개만 존재 할 경우 동작 유닛의 활용도가 크게 저하된다. 상기된 문제점을 해결하기 위해 제시된 것

이 SMT구조이다[3]. 이 SMT 구조는 ILP 와 TLP 의 특성을 최대한 활용함으로 성능의 향상을 목표로 하고 있다. 하지만 현재까지 제안된 SMT 구조를 구현하기 위해선 하드웨어 비용이 적지 않을 것으로 예상되어지고 있다. 그래서 본 연구실에서는 최소한의 하드웨어 비용으로 SMT 구조를 구현하고자 in-order issue and completion 형태의 프로세서 구조를 연구하고, 이를 이용한 네트워크 마이크로프로세서로의 전환도 연구하고 있다. 연구되어진 마이크로프로세서는 ARM ISA 구조를 기반으로 하고 있는 슈퍼스칼라 구조이며 SMT 구조에 알맞게 약간의 변형이 이루어져 있다. 최대 4개의 스레드를 지원할 경우의 프로세서는 다음 그림1과 같다.

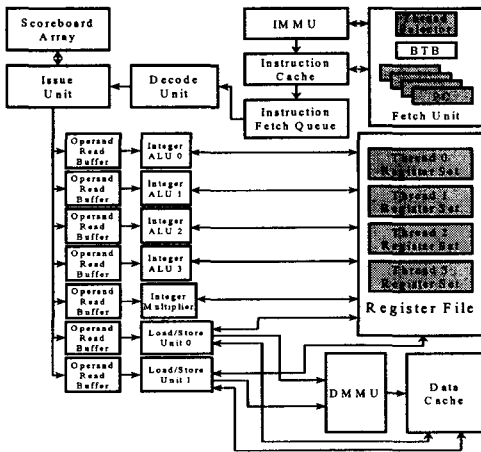


그림 1. 4개의 스레드를 지원하는 프로세서의 구조

그림 1 에서처럼 명령어 캐쉬 읽기 포트, 페치 큐, 디코드 유닛, 레지스터 파일의 읽기/쓰기 포트, 명령어 윈도우, 동작 유닛, 데이터 캐쉬, 결과 버스 등의 자원은 스레드 간에 동적으로 공유되도록 하였다.

연구된 SMT 마이크로프로세서의 파이프라인 구조를 표현하면 다음과 같으며 각각의 상태를 개략적으로 설명한다.

- 선택 단계(S) : 매 사이클마다 어떤 스레드들로부터 명령어들을 페치 할지를 결정한다.
- 페치 단계(F) : S 단계에서 선택된 스레드들로부터 명령어들을 페치 하여 페치 큐에 저장한다.
- 디코드 단계(D) : 명령어들이 해석된다.
- 이슈 단계(I) : 종속 문제 또는 자원 충돌 문제가 없는 명령어들이 동작 유닛들로 보내어 진다.
- 읽기 단계(R) : 명령어 연산자들을 레지스터 파일에서 읽어 오며, 다중 사이클 명령의 경우에는 상태 머

신을 통해 동작 유닛의 제어를 시작한다.

- 실행 단계(E) : 동작 유닛에서 명령어들을 실행하고, 메모리 주소 등을 계산하며, 그 결과는 결과 버퍼에 저장한다.
- 메모리 단계(M) : 해당 메모리 주소값을 통해 데이터를 읽거나 쓴다.
- 기록 단계(W) : 실행 결과가 순서대로 레지스터 파일에 기록된다.

III. SMT ALU

3.1 SMT ALU의 특징

본 논문의 ALU가 기존의 ALU와 다른 특징은 다음과 같다. 첫째, 기존의 ALU가 하나의 레지스터 파일과 오퍼랜드를 주고 받는 것과 달리 각각의 스레드식별자를 가진 레지스터 세트들과 데이터 호출하는 구조를 가진다. 둘째, 레지스터 파일과의 오퍼랜드 호출 지연 시간으로 인해 새롭게 파이프라인된 구조를 가진다. 셋째, 데이터 바이패싱이 가능한 구조를 가진다. 넷째, 스레드 ID와 순서비트의 비교를 통한 선택적 플러쉬 기능을 지원한다.

3.2 SMT ALU의 전체구조

본 논문에서 제시하는 SMT에 적합한 ALU의 전체 구조는 크게 3단으로 구성된다. 첫째, 레지스터 파일 혹은 바이패싱된 데이터를 소스 오퍼랜드로 입력받는 R(Read)단이 있다. 둘째, 입력받은 오퍼랜드들을 연산 수행하는 E(Execution)단이 존재한다. 마지막으로 연산 수행하는 M(Memory)단으로 나누어진다. 그림 2는 SMT ALU의 전체구조를 나타낸다.

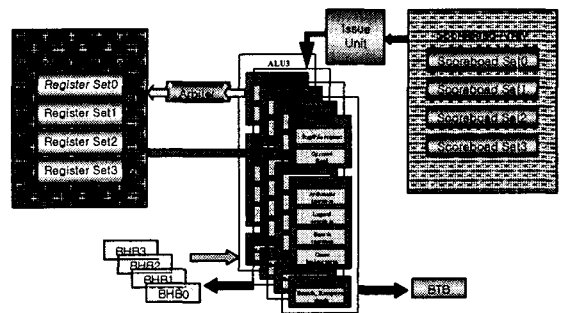


그림 2. SMT ALU의 전체구조

3.2 R(Read) 단계

ALU 파이프라인의 첫 번째 단계로서, 오퍼랜드를 읽어 오는 단계이고, 멀티 스레딩(multi-threading)이 가능해야 한다. 해당 스레드의 개수만큼 소스 오퍼랜드들이 필요하다. 그러므로 스레드 개수만큼의 레지스터 파일 세트들과 디펜던시 및 바이패싱의 정보를 담고 있는 스코어보드 어레이들이 스레드의 개수만큼 존재해야 한다.

ALU의 R단계에서는 간단한 명령어의 디코더가 존재하고, 이를 통해서 레지스터 번호와 레지스터 인에이블 신호를 추출한다. 추출된 신호를 레지스터 파일에 전송하여 레지스터 파일에서는 해당 레지스터 값을 ALU로 송출한다. 송출된 소스 오퍼랜드들은 오퍼랜드 버퍼에 저장된다. 이때, 그림 3에서 보는 바와 같이 해당 오퍼랜드를 입력받을 때 지연시간(12.9ns)이 커지는 문제가 발생되어 목표 속도인 100MHz를 만족시키지 못하는 관계로 R단을 RR0, RR1 두개의 파이프라인으로 나눌 필요성이 제기 되었다. 레지스터 파일과의 오퍼랜드 전송 지연시간의 문제로 새롭게 파이프라인 되었다. 그림 4는 ALU R 단계의 기본 블록을 나타낸다. 이와 함께, 스코어보드 어레이에서는 바이패싱 정보를 ALU의 R단으로 보내준다. 입력받은 바이패싱 정보는 해당 기능 유닛정보와 파이프라인 위치 정보를 가리킨다.

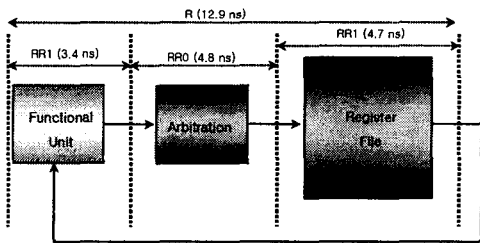


그림 3. 레지스터 호출에 대한 지연시간

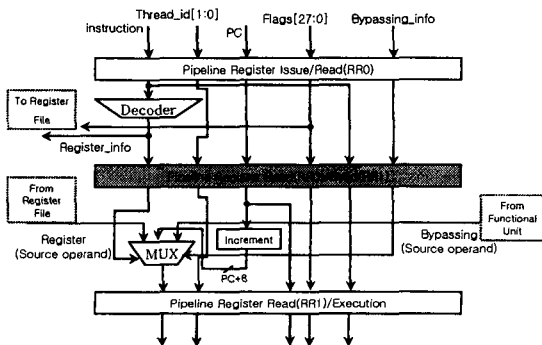


그림 4. SMT ALU R단계의 기본 블록

3.3 E(Execution) 단계

ALU의 E단계는 ALU 명령어들의 연산을 수행하는 단계이다. 본 논문에서 채택된 명령어 세트 구조에 따라 크게 나누어 컨디션 체크, 쉬프터 연산, 산술 연산, 논리 연산, 분기 명령어 연산, 선행 제로 카운터 연산, 그리고 확장DSP명령어 연산 등으로 나누어진다.

채택된 RISC ISA 구조에 따라 컨디션 체크의 수행은 레지스터 파일 내의 CPSR의 조건 플래그들의 비교 연산을 통해 이루어지고 조건부 연산 수행으로 인한 분기 명령어 대체와 분기 명령어 수행에 따른 분기 페널티의 감소의 효과를 가질 수 있다. 쉬프터는 2단 배럴 쉬프터를 사용하였고 ALU와 쉬프터의 시리얼한 연결구조로 인한 고속의 데이터 처리 효과를 가질 수 있는 구조로 설계되었다[4]. 오버플로우나 언더플로우 발생시 연산 결과를 saturation 처리로 확장 DSP 연산을 수행한다. 그림 5는 E단계의 구조를 보여 준다.

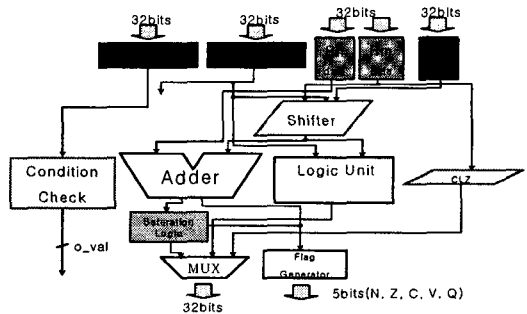


그림 5. SMT ALU의 E단계의 기본 블록

3.4 M(Memory) 단계

ALU의 M단계에서는 크게 분기 명령어 예측 검사와 예외 처리 그리고 CPSR, SPSR의 업데이트를 수행한다. 분기 명령어 예측 검사는 분기 예측 히트/미스 여부로 판단한다. 미스일 경우 BTB와 BHB에 갱신을 위해서 신호를 보낸다. 해당 스레드의 명령어들만 파이프라인 상에서 모두 플러쉬 된다. 그림 6은 분기 예측 오류의 검사 과정을 보여준다.

M단계에서는 분기 명령어 예측 오류나 예외 처리시의 명령어 플러쉬 기능을 수행한다. 그림 7은 플러쉬 되는 과정을 보여준다. ALU0유닛에서 스레드 식별자가 01이고, 순서비트가 0일 경우에서 분기 예측 오류가 발견되었다고 가정하면, ALU0에서는 해당 명령어의 플러쉬 되는 스레드 식별자와 순서비트를 발생한다. 나머지 기능유닛으로 신호가 전달되고 각각의 유닛에서 스레드 식별자와 순서비트를 비교하게 된다.

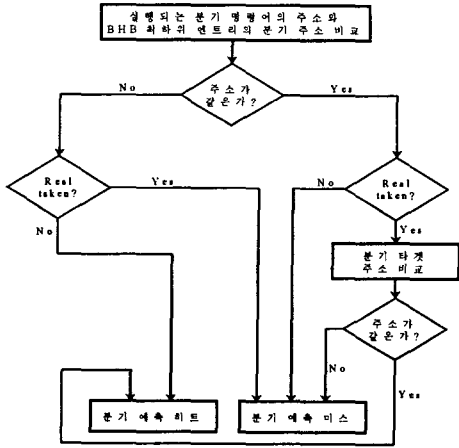


그림 6. 분기 예측 오류의 검사 과정

ALU2의 M단에서 같은 스레드 식별자를 가지고 플러쉬 되는 명령어보다 순서비트가 후순위가 된다고 가정한다. 이때 각각의 해당 명령어들 즉, 같은 스레드 식별자를 가지고 후순위 비트를 가지는 명령어들은 플러쉬가 된다. 그리고 ALU1의 M단에서 또다른 예외 처리신호가 발생한다면, ALU3의 R단에 존재하는 같은 스레드를 가지는 명령어는 플러쉬 됨을 알 수 있다. 여기서 순서비트는 비교하지 않아도 되는데 이것은 in order issue and completion의 정책으로 인해 같은 스레드의 R단의 명령어는 당연히 M단계의 명령어보다 후순위 명령어임을 알 수 있기 때문이다. 그리고 각각의 ALU의 다른 단계에 있는 명령어는 순서대로 수행된다.

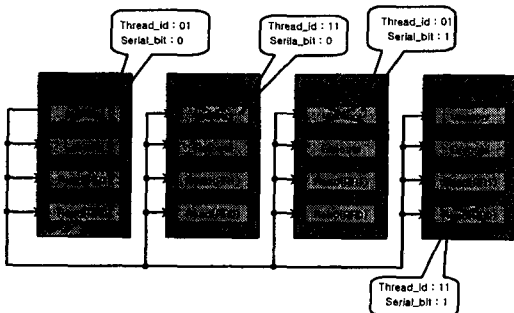


그림 7. 명령어 플러쉬

IV. 시뮬레이션 및 합성

본 논문의 SMT ALU는 Verilog HDL을 사용하여 구현되었고, 기능 검증을 위해 각각의 기능에 대해서

각각 1000개의 테스트 벡터를 사용하여 각 기능 블록의 동작을 확인하고 기능 수준과 게이트 수준에서 시뮬레이션을 수행하여 검증하였다. 본 논문에서 사용된 표준 셀 라이브러리(standard cell library)는 삼성전자의 0.35 μ m 공정의 std90 라이브러리로 합성하였다. 합성 툴은 Synopsys Design Analyzer를 사용하였다. 최악 조건 85 $^{\circ}$ C, 3.0V에서 합성 결과 임계 경로를 통해 지연되는 시간이 8.5ns가 소요되고 동작 주파수는 약 118MHz로 동작한다.

V. 결론

본 논문에서는 SMT 마이크로프로세서에 적합한 ALU를 설계하였다. SMT의 특징인 다중스레딩 환경에서 SMT ALU의 특징인 데이터 바이패싱과 분기명령어 실행과 예측 오류 처리가 가능하다.

SMT ALU는 Verilog HDL로 하드웨어를 설계하여, 0.35 μ m 표준 셀 라이브러리를 이용하여 합성하였다. 최악 조건에서 약 118MHz, 표준 조건에서는 145MHz의 동작 주파수를 보인다. 설계된 ALU의 속도나 면적 보다는 멀티 스레딩 지원을 위한 데이터 바이패싱, 명령어 플러쉬등의 기능에 역점을 두고 설계되었다. 따라서 설계된 ALU는 고성능 프로세서에 적합하다 할 수 있다.

참고문헌

- [1] Nikitas Alexandridis, Design of Microprocessor Based Systems, Prentice Hall. pp 246~312, 1993.
- [2] Suan J. Eggers, Joel S. Emer, Henry M. Levy, Jack L. Lo, Rebecca L. Stamm, Dean M. Tullsen, "Simultaneous Multi-threading : A Platform for Next-Generation Processors", IEEE Micro, p.p 12~19, September/October 1997.
- [3] 문병인, "순차적 SMT 구조 및 그룹화 방안"에 관한 연구" 연세대학교 박사학위 졸업논문, 2001
- [4] 이용석, "고성능 마이크로프로세서 ALU와 레지스터 파일의 구조," 비디오 및 온라인 강좌 시리즈, 2000, <http://mpu.yonsei.ac.kr/Lecture/nrl/ALURegisterFile.htm>