

저전력 기법을 적용한 ARM7 마이크로프로세서의 FPGA 구현 및 측정

김재우, 김영훈, 오민석, 남기훈, 이광엽
서경대학교 컴퓨터공학과
전화 : 02-940-7240 / 핸드폰 : 018-339-6949

FPGA Implementation and Measurement of ARM7 Microprocessor based on a Low-Power Method

Jae-Woo Kim, Young-Hun Kim, Min-Seok Oh, Ki-Hun Nam, Kwang-Youb Lee
Dept. of Computer Engineering, Seokyeong University
E-mail : kim9128@dreamwiz.com

Abstract

본 논문에서는 저전력 마이크로프로세서를 개발하기 위해 ARM7 마이크로프로세서와 명령어 호환을 갖는 32비트 RISC 구조의 마이크로프로세서를 설계하였다. 저전력 ARM7 마이크로프로세서 IP 구현을 위하여 새로운 정수 나눗셈 명령어를 정의하고 이를 적용하는 회로를 설계하여 제수가 피제수보다 큰 경우 6.4nW, 그 이외의 경우에는 76.5 nW를 소모하여 기존의 방법보다 140 ~ 860% 까지 개선되었음을 측정하였다. 또한 Multi-cycle 명령어 발생시 Prefetch에 의한 전력 소모를 줄이기 위하여 명령어의 condition code를 미리 결정함으로써 50%의 prefetch 동작 횟수를 줄였다. 그 결과 저전력 파이프라인의 경우에는 1.943mW/ 1MHz의 소비 전력이 측정되었다.

I. 서론

최근 IP(Intellectual Property)를 이용한 SoC(System On a Chip) 설계가 반도체 설계의 새로운 방안으로 대두되면서 핵심적인 Core IP 개발 및 이를 이용한 SoC 설계가 새로운 산업으로 등장하고 있으며, 휴대폰 및 PDA 등의 모바일 제품들의 급격한 수요로 인하여 SoC를 통한 모바일 임베디드 시스템(Embedded System) 설계 시 제한된 배터리 용량에서

보다 긴 사용시간을 갖는 저전력 Core IP의 사용이 필수적이다. 현재 SoC 설계에 있어 핵심적인 Core IP인 ARM7 TDMI 마이크로프로세서는 타 마이크로프로세서 제품에 비해 보다 적은 전력을 소비하나, 연산 블록의 구조와 외부 메모리로부터 명령어 인출 과정에 의한 전력 소모를 더 줄일 수 있다.

본 논문은 ARM7 TDMI 마이크로프로세서의 전력 소모를 기존보다 더 줄이기 위하여, 기존 파이프라인 동작 중에서 외부메모리로부터 명령어를 인출하는 단계인 IF단계를 재 설계하였으며, 긴 수행 시간으로 인해 많은 전력을 소모하는 기존의 나눗셈 연산 과정을 보다 짧은 시간에 수행하여 전력 소모를 줄이기 위하여 기존의 연산 블록을 재 설계하였다. 본 논문이 제안한 저전력 ARM7 마이크로프로세서는 Xilinx사의 Vertex-e1000 FPGA 장치에서 수행을 검증하였으며, Xilinx사의 전력 측정 툴인 XPOWER를 이용하여 동적 파워 소모(dynamic power consumption)를 측정하여, 기존의 ARM7 마이크로프로세서와 비교, 검증하였다.

II. 본론

2.1 저전력 명령어 인출

마이크로프로세서는 메모리로부터 데이터와 명령어를 전달받는다. 이 때 마이크로프로세서는 명령어 또는 데이터의 주소와 제어신호를 메모리에 전달한다^[1].

※ 본 연구는 IDEC과 IT_SoC 사업단 지원으로 수행되었습니다.

이 일련의 과정은 메모리보다 고속으로 동작하는 마이크로프로세서 입장에서는 수행에 많은 부담이 될 뿐만 아니라 많은 전력이 소비된다. 특히 수행 시 매 순간 명령어를 메모리로부터 전달 받아야하므로 이는 마이크로프로세서의 소비 전력을 더욱 크게 하는 원인이 된다. 그러므로 마이크로프로세서는 명령어가 필요치 않을 때 메모리로부터 명령어를 전달받지 않는다면 소비 전력을 줄일 수 있다. ARM7 마이크로프로세서는 IF(명령어 인출), DE(명령어 해석), EX(명령어 수행) 단계를 갖는 3 단계 파이프라인 구조로 파이프라인 동작 및 메모리와 데이터 전송은 그림 1과 같다^[1].

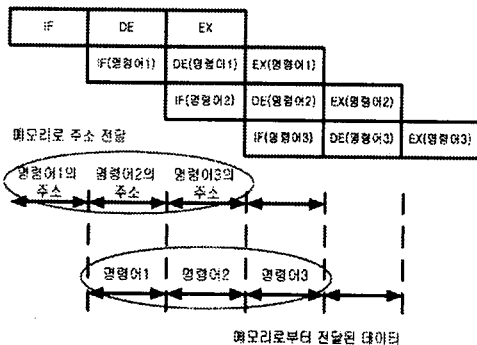


그림 1. ARM7 마이크로프로세서의 파이프라인 동작

명령어1, 2, 3은 단일 사이클 수행 명령어이며, DE단계와 EX단계를 거쳐 파이프라인 처리되어서 매 EX단계마다 결과를 발생한다. 만일 명령어1이 다중 사이클 수행을 갖는 분기 명령어일 경우 그림 1의 정상적인 파이프라인 동작은 그림 2와 같이 변형된다. 분기 명령어인 명령어1은 3사이클 수행을 갖는 다중 사이클 수행 명령어로 3개의 DE단계와 EX단계를 갖는다. 명령어1의 수행동작은 명령어1의 분기 타겟 명령어인 명령어#으로 분기하는 것이며 명령어의 수행 순서가 명령어1 -> 명령어# -> 명령어#+1이 된다. 그러므로 명령어2와 명령어3은 IF단계에서 메모리로부터 명령어를 인출하였어도 분기 명령어인 명령어1에 의해서 수행을 하지 않는다^[1,2]. 만일 다중 수행 사이클을 갖는 분기 명령어 실행 시 수행되지 않는 다음 명령어와 그 다음 명령어인 명령어2, 3을 메모리로부터 인출하지 않으면 ARM7 마이크로프로세서가 메모리로 접근하기 위해 소모되는 전력을 줄일 수 있다.

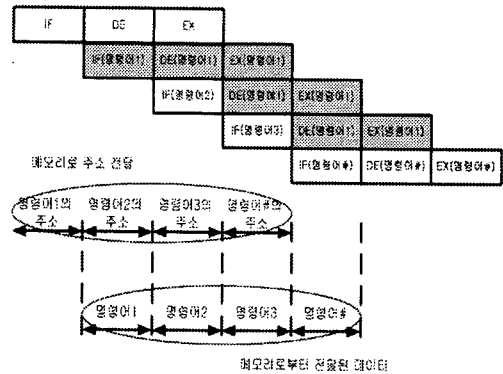


그림 2. ARM7 마이크로프로세서의 분기 명령어 수행 파이프라인 동작

모든 ARM 명령어는 조건부 실행을 하며, 분기 명령어도 예외는 아니다. ARM 명령어 중 조건을 발생하는 명령어는 EX단계에서 조건 플래그들을 발생하여 상태 레지스터의 조건 부분에 저장하며, 모든 ARM 명령어는 명령어가 포함하고 있는 조건과 상태 레지스터의 조건을 DE 단계에서 비교하여 명령어 수행 여부를 결정한다. 그림 2의 분기 명령어인 명령어1이 DE 단계에서 수행하는 조건 비교에 만족하지 못하여 수행되지 못할 경우 그림 1과 같이 명령어2와 명령어3 수행되어 정상적인 파이프라인 동작을 한다. 본 논문에서는 ARM7 저전력 파이프라인에서는 메모리로부터 불필요한 명령어 인출로 인한 전력 소모를 줄이기 위해 다음과 같은 항목을 제안하였다.

1. DE단계에서 수행하는 명령어 판별 동작을 분기 명령어 경우 IF단계에서 수행한다.
2. DE단계에서 수행하는 명령어 수행 조건 비교를 분기 명령어 경우 IF단계에서 수행한다.
3. 분기 명령어 바로 이전 명령어가 조건 플래그를 발생하는 명령어인지를 확인한다.

그림 3은 제안하는 명령어 수행 조건 비교 동작의 파이프라인을 나타내며, 현재 ARM7 마이크로프로세서의 분기 명령어 동작과 같이 불필요한 2개의 명령어를 메모리로부터 인출하는 것과 달리 전력 소모를 줄일 수 있다.

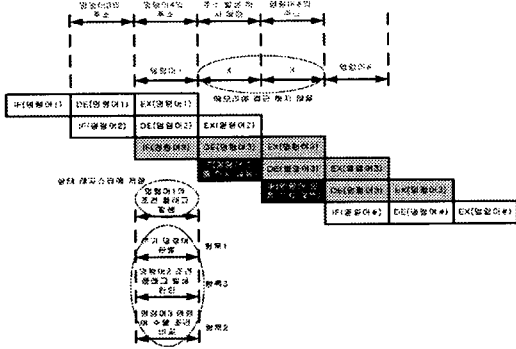


그림 4. 제안하는 저전력 명령어 인출 동작

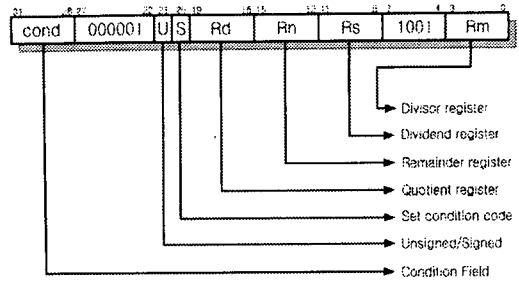


그림 4. 제안된 정수 나눗셈 명령어 형식

2.2 나눗셈 명령어 설계

ARM7 마이크로프로세서는 정수 처리 코어(Integer Core)로 다양한 정수 산술, 논리연산 기능을 갖고 있으나, 사칙연산 중 하나인 나눗셈 연산을 처리하기 위한 별도의 명령어와 데이터 패스를 갖고 있지 않다^[1,2]. 이는 대부분의 알고리즘은 나눗셈보다는 덧셈, 뺄셈, 또는 곱셈을 더 빈번하게 사용하고, 32bit 나눗셈을 수행하는데 요구되는 하드웨어 유닛은 복잡도와 면적으로 인해 높은 비용을 필요하기 때문이다. 그래서 현재 ARM7 마이크로프로세서에서는 소프트웨어 루틴들이 반복 알고리즘을 사용하여 정수 나눗셈을 수행한다^[3]. 그러나 정수 나눗셈 연산은 임베디드 시스템에서 데이터 처리와 데이터 구조 구현 시 빈번히 사용되며, 이를 처리위해 소프트웨어 루틴의 수행 시 많은 반복 연산으로 인해 긴 수행 시간을 갖고, 이는 많은 전력 소모를 발생한다. 본 논문에서는 ARM7 마이크로프로세서의 연산기능 중 구현되지 않은 정수 나눗셈 연산 기능을 제안하였다. 이를 위해 이를 위해 부호 없는 정수 나눗셈 명령어인 'UDIV' 명령어와 부호 있는 정수 나눗셈 명령어인 'SDIV' 명령어를 새로 정의하였으며, 새로 정의된 정수 나눗셈 연산 명령어 UDIV는 부호 없는 정수 나눗셈 연산과 부호 없는 모듈러 연산을 동시에 수행하고, SDIV는 부호 있는 나눗셈 연산과 부호 있는 모듈러 연산을 동시에 수행하며, 수행 동작은 $Rd = Rs/Rm$, $Rn = Rs \text{ mod } Rm$ 이다. S bit의 상태에 따라 연산 결과의 상태(Z flag : 연산 결과가 0, N flag : 연산 결과가 음수)를 상태 레지스터인 CPSR(Current Processor Status Register)에 저장할 수 있다. 제안된 정수 나눗셈 연산 명령어의 형식은 그림 4와 같으며, 제안한 정수 나눗셈 명령어의 수행을 위한 데이터 패스는 radix-2 비복원 알고리즘을 적용하여 설계하였다^[4,5].

비복원 나눗셈 알고리즘을 수행하기 위해 필요한 연산 기능은 현재 ARM7 TDMI 마이크로프로세서의 데이터 패스가 대부분 갖고 있다. 몫 결정 방법은 부분 나머지의 부호와 제수의 부호를 이용하여 몫을 결정할 수 있으며, 새로운 부분 나머지를 구하는 방법 역시 ALU의 덧셈, 뺄셈 연산과 이용할 수 있다. 그림 5는 비 복원 나눗셈 알고리즘 수행을 위해 제안된 ALU 주변의 데이터 패스이다. 어둡게 표시된 부분이 추가된 하드웨어 유닛이며, 32bit 멀티플렉서 3개, 1bit 래치 2개, 64bit 쉬프트 레지스터 1개로 구성된다.

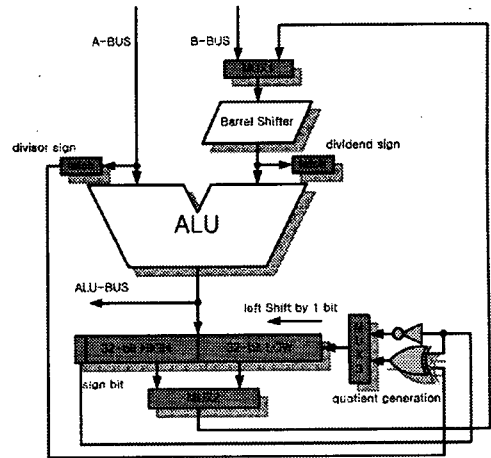


그림 5. 비복원 나눗셈 알고리즘을 적용한 데이터 패스

III. 전력 측정

3.1 ARM7 마이크로프로세서 전력 측정

ARM7 마이크로프로세서의 전력을 측정하기 위하여 사용한 테스트 벡터는 1부터 10까지 덧셈을 하는 테스트 벡터와 NOP 명령어로 구성된 테스트 벡터 2가지를

ARM7 마이크로프로세서에 적용하여 측정하였다. 그림 6은 두가지 테스트 벡터를 비교한 그래프이다.

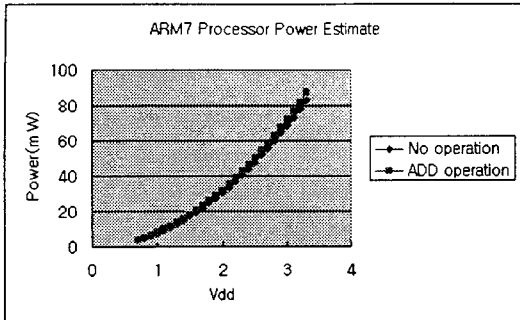


그림 7. 덧셈 명령어와 NOP 명령어의 전력 추정 비교

3.2 나눗셈 명령어를 포함한 ARM7 마이크로프로세서 전력 측정

기존 ARM7 마이크로프로세서의 전력은 클럭 사이클당 2.053mW/1MHz가 소요되는 반면 나눗셈 명령어를 포함한 ARM7의 전력은 2.125mW/1MHz가 소요된다. 클럭 사이클 당 전력이 늘어난 이유는 기존 ARM7에 나눗셈 명령어를 수행할 수 있도록 몇 가지의 회로가 추가돼서 전력이 기존 ARM7 마이크로프로세서에 비해 늘어났다. 그러나 표 1에서 보듯이 나눗셈 명령이 ARM7 마이크로프로세서에서 수행되는 경우에는 많은 전력 절감 효과를 나타내는 것을 알 수가 있다.

표 1. 나눗셈 명령어가 포함된 ARM7의 전력 소모

경우	소프트웨어 루틴의 부호 있는 정수 나눗셈 처리			DIV 명령어의 부호 있는 정수 나눗셈 처리			
	수행 사이클 수	수행 명령어 수	전력 소비	수행 사이클 수	수행 명령어 수	전력 소비	전력 개선
계수=0	소프트웨어 인터럽트 발생			소프트웨어 인터럽트 발생			
계수 > 피계수	27	18	55.4nW	3	1	6.4nW	860%
그 외의 경우	52-234	44	106.8-408.4 nW	36	1	76.5nW	140-630%

3.3 저 전력 명령어 인출의 전력 측정

ARM7 마이크로프로세서의 저 전력 파이프라인에서는 분기 명령어시 다음 명령어와 그 다음 명령어를 fetch하지 않음으로써 메모리 액세스를 줄여 소비 전력을 줄이는 방법이다. 저 전력 파이프라인을 적용하지 않은 ARM7 마이크로프로세서의 경우에는 2.053mW/1MHz의 소비전력을 소비하는 반면 분기 명령어 수행시 다음 두개의 명령어를 fetch하지 않는 저 전력 파이프라인의 경우에는 1.943mW/1MHz의 소비 전력을 소모한다.

IV. 결론

본 논문에서는 저 전력 SoC 시스템에서 활용이 가능한 저 전력 마이크로프로세서를 개발하기 위해 ARM7 마이크로프로세서와 명령어 호환을 갖는 32비트 RISC 구조의 마이크로프로세서를 설계 하였다. 저 전력 ARM7 마이크로프로세서 IP 구현을 위하여 새로운 정수 나눗셈 명령어를 정의하고 이를 적용하는 회로를 설계하여 전력 소모 측면에서는 계수가 피계수보다 큰 경우 6.4nW, 그 이외의 경우에는 76.5 nW를 소모하여 기존의 방법보다 140 ~ 860% 까지 개선되었음을 측정하였다. 또한 Multi-cycle 명령어 발생시 Prefetch에 의한 전력 소모를 줄이기 위하여 명령어의 condition code를 미리 결정함으로써 50%의 prefetch 동작 횟수를 줄였다. 그 결과 저전력 파이프라인의 경우에는 1.943mW/ 1MHz의 소비 전력이 측정되었다.

참고문헌

- [1] Steve Furber, "ARM System on a chip Architecture, Addison-Wesley", pp. 75-76, 1996
- [2] ARM7 TDMI Data Sheet (ARM DDI0029E), Advanced RISC Machines Ltd(ARM), 1995
- [3] Programming Techniques (ARM DUI0021A), Advanced RISC Machines Ltd (ARM), 1995
- [4] Behrooz Parhami, "Computer Arithmetic, Algorithms and Hardwarw Design", Oxford University Press, pp. 128-211, 2000
- [5] Israel Koren, "Computer Arithmetic Algorithms", John Wiley & Sons. Inc, pp. 35-40, 1978