

ARM7 호환 32-Bit RISC Microprocessor 설계¹

이기호*, 유영재, 김기민, 강용호, 송호준, 이철훈
충남대학교 컴퓨터공학과

A Design of the ARM7-Compatible 32-Bit RISC Microprocessor

Ki-Ho Lee*, Young-Jae Yoo, Ki-Min Kim, Yong-Ho Kang, Ho-Jun Song, Cheol-Hoon Lee
Dept. of Computer Engineering, Chungnam National University

요 약

본 논문에서는 RISC Microprocessor Core 설계에 대한 기반 기술을 확립하며, GPS(Global Positioning System)같은 Embedded 시스템 등에서 주로 사용되어 지고 있는 ARM사의 ARM7 CPU와 이진 호환이 가능한 Microprocessor를 설계하고자 하였다. 이를 위하여 RISC Microprocessor의 기본적인 구조를 바탕으로 하여 ARM7 CPU와의 호환을 위해서 ARM7 CPU의 명령어들이 주어진 Clock안에 수행될 수 있도록 설계를 하였고 여러 모듈들을 원활히 공유할 수 있도록 내부에 공유 버스를 설계하였다.

설계를 위해서 Verilog-HDL(Hardware Description Language)을 사용하였으며, Microprocessor를 기술하는데 있어서 Behavioral 구조와 RTL(Register Transfer Level) 구조를 혼합하여 사용하였다. 설계된 Microprocessor의 동작은 면적과 타이밍의 최적화를 거친 후 Cwaves 툴을 사용하여 실질적인 ARM7의 명령어들을 수행하면서 검증하였다.

1. 서론

컴퓨터 산업의 기하급수적 발전과 반도체 기술의 발달로 Microprocessor는 현대 디지털 산업 사회에서 없어서는 안될 요소로 자리잡았으며, 성능 향상과 고집적화에 따른 요구와 VLSI(Very Large Scaled Integration)기술의 비약적인 발달로 인하여 다양한 기능과 뛰어난 성능을 소형의 단일 칩에 모두 함축시킬 수 있게 되었으며 앞으로 이러한 요구 조건의 필요성으로 인해 더욱 발전해 나갈 것이다[8].

Microprocessor의 구조에는 RISC(Reduced Instruction Set Computer), CISC(Complexed Instruction Set Computer), 그리고 RISC와 CISC의 혼합 구조가 있으나 Embedded 시스템에서는 chip의 면적과 간단한 기능 요구에 따라 주로 RISC Microprocessor를 사용하고 있고, ARM사의 ARM 계열 chip들 역시 RISC 구조를 바탕으로 설계되었기 때문에 호환 chip의 설계를 목적으로 하는 본 논문에서도 역시 RISC 구조가 바탕이 되었다.

본 연구에서는 우리가 알고 있는 기본적인 Microprocessor의 구조에 덧붙여 RISC Microprocessor의 구조를 파악하고, 이미 상용화 되어 여러 제품들에 사용되어지고 있는 32bit Microprocessor와 이진 호환이 가능한 Microprocessor를 자체 설계하여 현재 국내에서 부족한 비메모리 반도체분야에서 Microprocessor 설계와 디지털 제어의 기본 기술을 다지며, 이미 상용화된 chip과 호환 가능한 Microprocessor를 설계함으로써 시간 및 인원, 그리고 경제적인 비용 부담을 줄이는데 중점을 두었다.

2. RISC Microprocessor 설계

2.1 일반적인 Microprocessor 구조

기본적인 Microprocessor의 구조를 살펴보면 먼저 명령어의 주소를 메모리로 보내는 부분(Address Register), 명령어를 데이터 버스로부터 읽어 들여 저장하는 부분(Instruction Register)이 필요하며, 읽어온 명령어를 해석하여 특정 신호를 분류해 내는 부분(Decoder), 해석이 된 명령어를 각 명령어 종류에 따라 실행하는 부분(ALU, Multiplier, Shifter등), 명령어 실행에 필요한 오퍼랜드와 실행 결과가 저장되는 부분(Register), 각 부분들을 서로 연

¹ 본 연구는 1997년도 학술진흥재단 대학부설 연구소 과제에 의하여 연구 되었음

결해주는 버스(Bus), 그리고 이 모든 부분들을 제어하는 부분(Controller)이 필요하다[3][4][7]

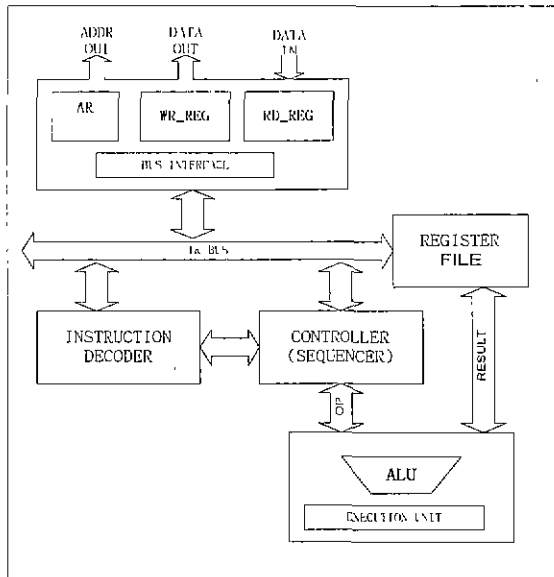


그림 1. 일반적인 Microprocessor 구조

그림 1은 일반적인 Microprocessor 구조를 간단히 나타낸 것이다.

2.2 ARM7 Microprocessor 구조와 특징

본 논문에서 연구한 관련 연구 대상으로 채택한 ARM7 Microprocessor의 구조를 살펴보면 다음과 같다. ARM7 Microprocessor는 RISC Microprocessor의 기본적인 특징과 구조를 모두 포함하고 있으며 여기에 더하여

- 고정된 32bit 명령어와 데이터
- 배럴 쉬프터(Barrel Shifter)
- 부스 승산기(Booth Multiplier)
- 31개의 레지스터 파일(6개의 레지스터 윈도우)
- 3단 파이프 라인
- Load, Store만 메모리 접근
- 26bit 명령어와 데이터 지원
- Big endian, Little Endian 지원

의 구조와 특징을 가지고 있다[1].

2.3 Microprocessor 설계

앞 절에서 기술된 RISC Microprocessor와 ARM7 cpu의 구조와 특징들을 조건으로 하여 설계에 임하였으며, 설계 과정을 보면 Verilog-HDL[2][5][6]을 사용하여 먼저 ARM7 Microprocessor의 구조와 특징에 맞추어 모듈들을 설계하

고, 다시 각 모듈간의 버스와 인터페이스를 설계하였으며, 각 모듈이 만들어질 때마다 동작을 검증한 후 전체 구조에 추가하는 방법을 택했다

각 모듈의 동작은 클럭의 positive edge에 의해 동기화되어 운용되도록 자원들을 설계, 구성하여 개개 모듈에서 발생하는 지연 시간들이 다음 명령어까지 누적되지 않도록 하였다. 단, ARM7 Microprocessor의 특징 중 하나인 26bit 명령어와 데이터 지원은 본 논문에서는 제외 하였다.

본 논문에서 계획했던 Microprocessor의 필요한 구성 요소들은

- Address register
- Register file
- Decoder
- Barrel shifter
- Booth multiplier
- ALU
- 3-stage Pipeline
- Sequential counter
- BUS controller
- Control Logic

들이며, Arithmetic Unit의 Adder는 향후 구현된 다음 chip의 빠른 응답과 실질적인 면적을 고려해서 gate 레벨에서 Carry Lookahead Tree Adder로 설계하였다.

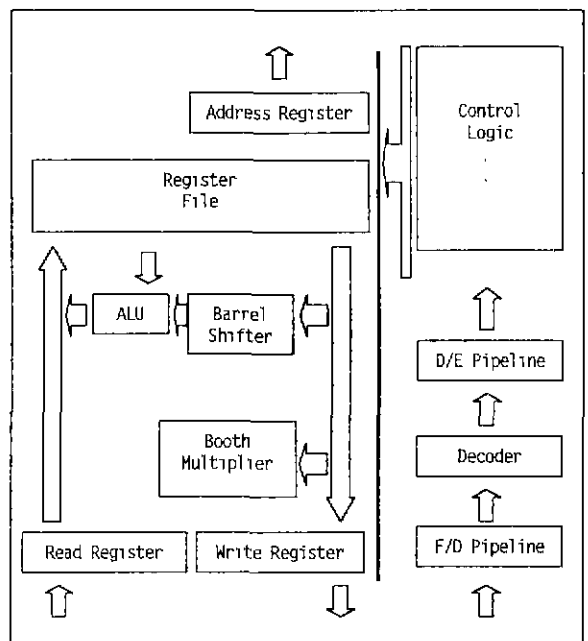


그림 2. 설계된 3 단 Pipeline RISC Microprocessor

각각의 모듈들을 주어진 명령어 사이클에 맞추어 동작 시키도록 제어부를 정의하였으며, 이 제어부 역시 클럭의 positive edge 에 의해 동기화 되어진다.

그림 2는 ARM7 chip 을 참고로 하여 설계한 Microprocessor 의 각 모듈들과 버스를 통한 상호 구성 관계를 간단히 블록화한 것이다.

2.4 Microprocessor 검증

본 연구에 의해서 설계된 RISC Microprocessor의 동작은 Synopsys사의 design_analyzer 틀의 면적과 타이밍 최적화를 거친 후 Cadence사의 Cwaves 볼을 사용하여 검증하였다.

검증 형식으로서 외부에서 연속적인 클럭(40MHz)을 공급하고 매 클럭마다 ARM7의 명령어들을 비트 형식으로 입력시켜주었을 때, 설계한 Microprocessor가 ARM7의 명령어 수행 사이클과 같은 사이클안에 정확히 동작함을 확인하였다.

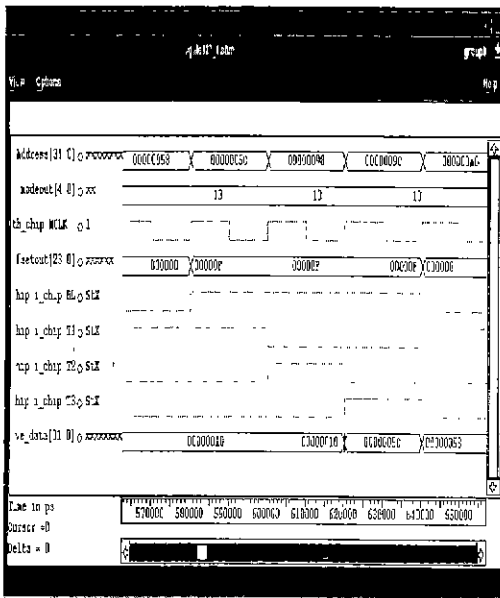


그림3 시뮬레이션 결과

그림 3은 Cwaves볼 상에서, 설계된 Microprocessor의 외부 명령어 입력으로 ARM7의 Branch 명령어를 비트 형식으로 주었을 경우, 그에 관련된 register의 값과 분기(Jump)할 주소를 정확하게 출력하는 시뮬레이션 결과를 보여준다. 각 신호를 살펴보면, ①은 address register 값의 변화를 나타내고,

②는 main clock, ③은 offset, ④는 Branch with Link 명령어의 시작과 종료, ⑤,⑥,⑦은 sequential count, ⑧은 return할 address를 저장한 register의 값을 나타낸다.

3. 결론 및 향후 과제

본 논문에서는 RISC Microprocessor의 구조에 대해서 알아보았고 향후 비메모리 반도체 분야에서 Microprocessor 설계의 기본 기술을 확립하고자 할 목적으로 상용화된 ARM7 CPU와 이진 호환이 가능한 Microprocessor를 설계하고 검증하였다.

본 논문의 향후 과제로는 지금까지 설계하고 검증한 Microprocessor를 FPGA(Field Programmable Gate Array)로 구현한 다음 실제 chip으로 생산해 내는 것이며, 현재 약 30,000 gate에 달하는 면적을 상용화된 제품의 수준과 비슷하게 줄여 나가는 것이다.

이에 더하여 GPS 시스템의 DSP(Digital Signal Processing)같은 부분의 기능을 Microprocessor에 혼합하여 효율적이고 경제적인 one-chip으로 구현하는 것이다. 이를 위해 외부 로직과의 기능적인 인터페이스와 실질적 chip의 면적을 최소화 할 수 있는 자원 공유 방법에 대한 연구가 이루어져야 할 것이다.

참고문헌

- [1] ARM Ltd, *ARM Architecture Reference Manual*, Prentice Hall, 1996
- [2] 박현철, *VHDL 회로설계와 응용*, 한성출판사, 1994
- [3] Kai Hwang, *Advanced Computer Architecture Parallelism, Scalability, Programmability*, McFraw-Hill, 1993
- [4] M Morris Mano, *COMPUTER SYSTEM ARCHITECTURE*, 3rd ed. Prentice Hall, 1992
- [5] Donald E.Thomas and Philip R Moorby, *The Verilog Hardware Description Language*, KLUWER ACADEMIC PUBLISHERS, 1991
- [6] CADENCE, *Verilog-XL Training Course*, Ver 3.3, 1990
- [7] D. Patterson and C Sequin, "A VLSI RISC," *IEEE computer*, 15(9), 1982
- [8] Michael J. Flynn, "Some Computer Organization and Their Effectiveness", *IEEE Trans on Computer* Vol. C21, N.9, Sep 1972