

8051 호환 마이크로컨트롤러의 설계

이 용 석, 이 성 원, 강 형 주, 김 진 석, 박 인 철
한국과학기술원 전자전산학과 전기 및 전자공학 전공
전화 : 042-869-4404 / 핸드폰 : 017-430-0324

Development of an 8051-compatible microcontroller

Yongseok Lee, Sungwon Lee, Hyeongju Kang, Jinseok Kim, Incheol Park
Dept. of Electrical Engineering and Computer Science, KAIST
E-mail: yslee@ics.kaist.ac.kr

Abstract

This paper presents the development of an 8051-compatible microcontroller. The 8051 microcontroller is one the most popular microcontroller used nowadays. All the features of the 8051, including peripherals, are implemented. The output of this work is a synthesizable RTL model that is readily available for a simple control unit in a more complex chip, such as an SOC. We put some important notes relating to the implementation of the 8051's features, including bit addressing, multiplication/division, etc.

I. 서론

8051 마이크로 컨트롤러 (이하 8051)는 1980년 인텔에 의하여 개발된 이래 제어를 비롯한 광범위한 분야에서 사용되어져 왔다. 복잡한 동작을 필요로 하지 않는 8 비트 마이크로 컨트롤러 계열에서는 독보적인 지지를 얻어왔으며 현재에도 그러한 추세는 계속되고 있다. 그러나 현재 시판되고 있는 8051 호환 마이크로 컨트롤러들은 초기의 설계 및 성능을 그대로 답습했기 때문에 빠른 속도가 요구되는 현대에는 적합하지 않으며, 따라서 호환성은 그대로 유지하면서

성능을 더욱 향상시킨 8051 마이크로 컨트롤러가 필요하게 되었다. 또한 최근에는 IP (intellectual property)들을 조합함으로써 SOC (system-on-chip)를 구현하는 것이 화두로 떠오르고 있으며 8051의 코어는 칩의 컨트롤러로 쓰일 수 있는 중요한 IP가 되었기 때문에 이의 독자적인 개발은 중요한 의미를 가진다.

본 논문은 인텔 8051 마이크로 컨트롤러 (MCS51[®])와 명령어 수준에서 완벽하게 호환하면서 독자적인 내부 구조를 가진 마이크로 컨트롤러 (이하 K8051)를 설계한 내용을 다룬다. 본 마이크로 컨트롤러는 CPU 뿐만 아니라 입출력 포트, 직렬 포트, 타이머 겸 카운터 등의 주변 로직까지 갖추고 있으며, MCS51[®]과 외부 핀 신호에서도 정확히 같은 동작을 보인다. 그러나 주변 로직 없이 embedded 용으로도 사용될 수 있도록 설계하였다. 연구의 산물을 ISS (instruction set simulator, 명령어 집합 시뮬레이터)와 완전히 합성할 수 있는 마이크로 컨트롤러의 HDL (hardware description language, 하드웨어 기술 언어)이다.

II. 8051의 설계

2.1 K8051의 구조

8051은 크게 CPU, 메모리, 주변 로직으로 이루어진다. 그림 1은 이의 전체적인 블록도이다.

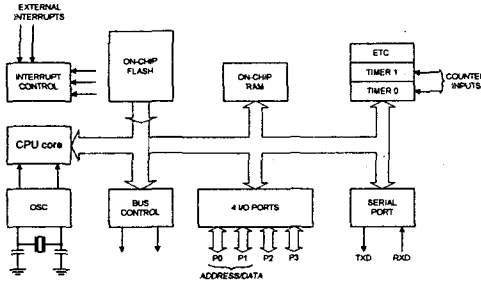


그림 1 The block diagram of 8051[1]

2.2 설계의 진행

K8051을 설계함에 있어서 가장 먼저 한 일은 설계 대상의 ISS (instruction set simulator)인 iss8051을 제작하는 것이었다. iss8051은 UNIX 환경에서 C로 구현하였으며 인터럽트를 포함한 8051 CPU core의 모든 동작을 시뮬레이션 할 수 있다. Text 기반에서의 실행과 GUI 기반에서의 실행을 모두 지원하며 gdb (GNU debugger)와 비슷한 환경을 제공한다.

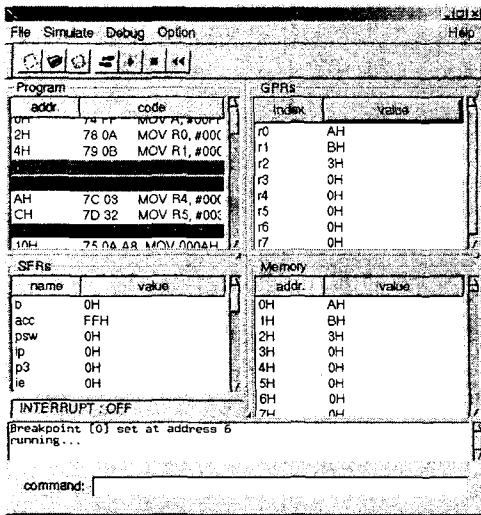


그림 2 GUI of iss8051

ISS를 만든 후에는 하드웨어의 각 부분을 설계하였고 이를 Verilog HDL로 기술한 후에 시뮬레이션과 FPGA 에뮬레이션으로 동작을 검증하였다.

2.3 컨트롤 부분

CPU에서 수행하는 명령어들은 1 바이트에서 3 바이트까지 가변적인 길이를 가지는 CISC 형태로 되어 있으며 각 명령어마다 수행 시간이 각각 다르기 때문에 CPU의 제어 로직은 마이크로 코드에 의한 방식으로 구현하였다.

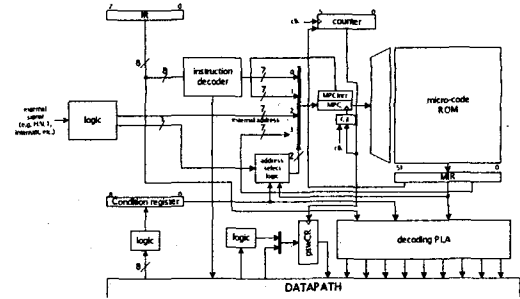


그림 3 Control part of K8051

명령어를 수행하는 첫 단계는 IR (instruction register)에 있는 값을 디코드하여 첫 마이크로 코드를 ROM으로부터 가져오는 것이다. 그 후에는 명령어의 수행이 끝났음을 알리는 end_of_instruction 신호가 뜰 때까지 마이크로 코드 ROM의 주소를 증가시키면서 다음 마이크로 코드를 가져온다. 명령어의 첫 바이트가 디코드되면 그 명령어의 길이 정보까지 함께 나오며 이것이 데이터패쓰의 PC (program counter)값을 얼마나 증가시킬 것인지를 결정한다.

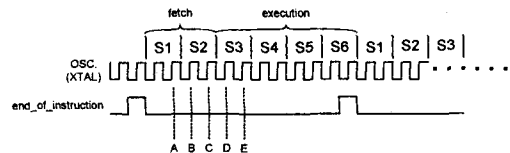


그림 4 Cycle behavior of K8051

위의 그림은 명령어를 수행할 때의 행태를 나타낸 것이다. K8051은 8051과 마찬가지로 두 개의 주기가 한 상태 (state)를 이루고 6 개의 상태가 모여 한 사이클을 이룬다. 그림 3에 예시된 것은 한 사이클 짜리 명령어를 수행하는 경우이다. 전 명령어의 수행이 끝났다는 의미로 end_of_instruction 신호가 발생하면 edge A에서 opcode가 IR에 들어온다. 이 opcode는 마이크로 코드 ROM의 해당 명령어에 대한 시작 주소로 디코드되어 edge B에서 MPC에 래치된다. MPC는 ROM에 직결되어 있으며 MPC 값에 해당하는 마이크로 코드는 edge C에서 MIR에 들어간다. 이 MIR의 값은 decoding PLA에 의해 제어 신호들로 디코드되어 데이터패스로 나가게 된다. 그리고 edge D에서는 MPC에 1이 증가된 값이 쓰여지게 되고 이에 해당하는 다음 마이크로 코드가 edge E에서 MIR에 래치된다.

Microcode ROM에는 12 bit 짜리 마이크로 코드가 814 개가 저장되어 있으며 총 1.2 Kbyte의 크기를 차지한다.

2.4 데이터패스

(1) Bit Addressing

8051은 비트 단위의 연산이 가능하다. 이를 위해서는 데이터를 비트 단위로 읽거나 쓸 수 있는 기능, 즉 bit addressing이 가능해야 한다. Bit addressing이 허용되는 부분은 SFR (special function register)의 일부와 데이터 메모리의 일부에 국한된다.

SFR과 MD (memory data register)의 bit-addressing은 다음과 같이 구현하였다.

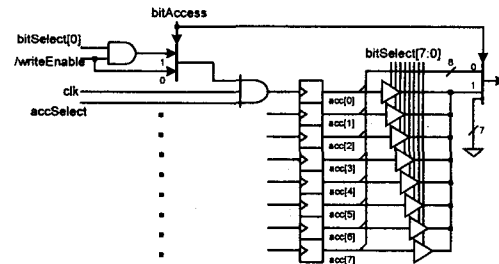


그림 5 Bit addressing of SFRs

(2) 곱셈과 나눗셈

K8051의 데이터패스에는 곱셈이나 나눗셈을 위한 하드웨어가 따로 존재하지 않기 때문에 ALU의 덧셈 뺄셈 기능과 데이터 버스만을 가지고 이들을 수행해야 한다. 곱셈, 나눗셈을 수행하는 데에는 쉬프트 연산이 필수적이기 때문에 ALU의 operand에 해당하는 tmp1 register와 tmp2 레지스터를 쉬프트 레지스터로 만들었다.

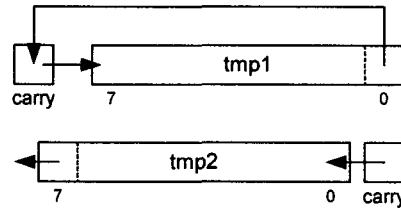


그림 6 Implementation of tmp1 & tmp2

그림 5에 도시한 바와 같이 tmp1은 carry를 포함한 left-rotation이 가능하고, tmp2는 carry에서 시작되는 right-shift가 가능하도록 설계했다. 곱셈과 나눗셈은 각각 가장 간단한 알고리즘인 add-and-shift 알고리즘과 shift-and-subtract 알고리즘 [2]을 사용하여 구현했으며 수행 시간은 각각 4 사이클과 6 사이클이 걸린다.

2.5 메모리

K8051은 single chip 뿐만 아니라 embedded 용으로도 쓰일 것을 목적으로 하기 때문에 메모리는 외부 메모리와 칩 상의 내부 메모리를 모

두 지원해야 한다.

원래의 8051은 프로그램 메모리와 데이터 메모리가 분리되어 있다. 프로그램 메모리는 4 KByte의 내부 메모리가 제공되며 64KByte까지의 외부 메모리를 사용할 수 있다. 데이터 메모리는 128 byte의 내부 메모리 또는 64 KByte까지의 외부 메모리를 사용할 수 있다. [1]

그러나 K8051은 이러한 8051의 메모리 구조를 지원함과 동시에 모든 메모리 주소 공간을 내부 메모리로 가정하고 사용할 수도 있게 되어 있다. 내부 메모리와 CPU의 인터페이스는 특별한 프로토콜이 필요하지 않은 SRAM을 가정하여 구현하였다.

III. 검증

K8051 HDL model의 검증은 두 단계로 진행했는데 첫째는 HDL의 동작을 C로 기술한 ISS의 동작과 비교하는 것이었다. 이것은 IPC (inter-process communication)를 사용해서 이루어졌다. ISS는 사이클 단위의 시뮬레이터가 아니라 명령어 단위의 시뮬레이터이기 때문에 명령어 수행 사이클보다 더 작은 시간 단위에 대한 동작은 검증하지 못한다. 따라서 ISS와의 비교 검증은 각 명령어의 수행이 끝나는 시점에서 수행의 결과만을 비교한다.

검증의 둘째 단계는 FPGA를 사용한 에뮬레이션이었다. HDL 모델을 합성한 결과를 FPGA에 매핑하고 이를 실제 회로와 함께 동작시켜 봄으로써 각 핀 신호와 내부 동작의 타이밍을 관찰할 수 있다. 그림 6은 검증 회로의 블록도이다.

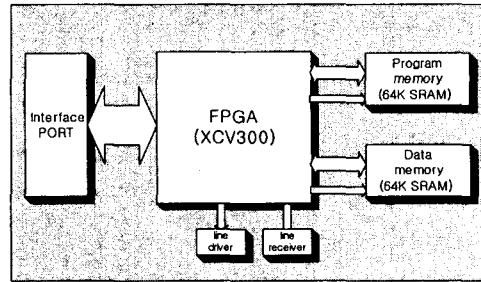


그림 7 Emulation Board Using FPGA

검증 회로는 PC로부터 프로그램과 데이터를 전송 받아 SRAM에 저장하고 FPGA에 매핑되어 있는 HDL 모델이 이것을 처리하여 직렬 포트나 입출력 포트를 통하여 PC에게 데이터를 전송함으로써 수행 결과를 확인할 수 있다.

IV. 결과

Synopsys Design compiler로 합성한 결과 총 게이트 수는 9600 개였다. 이는 현대 EML 0.18 μm CMOS 공정으로 합성할 경우 약 0.31 mm^2 의 면적을 차지할 것으로 예상된다.

본 연구에서는 인텔의 8051과 명령어 집합 및 핀 신호 수준에서 완전하게 호환하는 마이크로 컨트롤러인 K8051을 개발하였다.

K8051은 그 자체가 하나의 칩으로서 동작할 수 있을 뿐만 아니라 각종 칩에 embedded controller로 사용될 수 있기 때문에 많은 응용 분야에 활용될 수 있을 것으로 사료된다.

참고문헌

[1] Flash Microcontroller Architectural Overview, ATMEL, 1997
 [2] David A. Patterson and John L. Hennessy, "Computer Architecture: A Quantitative Approach," Morgan Kaufmann Publishers, Inc., 1995