ISSN: 1226-7244 (Print)
ISSN: 2288-243X (Online)
논문번호 15-04-24
166

j.inst.Korean.electr.electron.eng.Vol.19,No.4,625~631,December 2015
http://dx.doi.org/10.7471/ikeee.2015.19.4.625

데이터 전송을 위한 전용 명령어 및 I/O 포트를 탑재한 8051 마이크로콘트롤러의 설계

Design of an 8051 Microcontroller With Application-Specific Instructions and I/O Ports for Data Transmission

김 지 혜*, 이 성 수*★

Jihye Kim*, Seongsoo Lee*★

Abstract

In this paper, an 8051 microcontroller with application-specific instructions and I/O ports for data transmission is designed. The designed microcontoller includes two UART ports and one SMBus port to control external devices and to transmit data with them. Application-specific instruction is developed and added to the instruction set to exploit these I/O ports. So the designed microcontroller can perform multi-device control and multi-byte transmission. Also, it can reduce the code size of the application program. Especially, the designed microcontroller does not stall and can execute other programs during data transmission, which significantly increases its efficiency. Synthesized in 0.18 um technology, the area overhead due to application-specific instructions was negligible. Operations of all instructions and I/O ports were verified to run correctly on a FPGA board.

요 약

본 논문에서는 데이터 전송을 위한 전용 명령어 및 I/O 포트를 탑재한 8051 마이크로콘트롤러를 설계하였다. 설계된 8051마이크로콘트롤러는 외부 디바이스를 제어하고 데이터를 주고받기 위해 2개의 UART 포트와 1개의 SMBus 포트를 탑재하였으며, 이들 포트를 운용하기 위한 전용 명령어를 개발하여 명령어 집합에 추가하였다. 이에 따라 여러 디바이스를 동시에 제어하고 데이터를 전송할 수 있으며, 응용 프로그램의 코드 크기도 줄일 수 있다. 특히, 다수의 디바이스와 데이터를 주고받는 동안에도 마이크로콘트롤러가 멈추지 않고 다른 프로그램을 수행할 수 있어서 동작 효율을 크게 향상할 수 있다. 0.18 um 공정에서 합성한 결과, 전용 명령어 추가로 인한 하드웨어 크기 증가는 무시할만하며, 모든 명령어 및 I/O 포트가 정상적으로 동작하는 것을 FPGA 보드에서 확인하였다.

Key words : microcontroller, serial interface, UART, SMBus, application-specific instruction

* School of Electronic Engineering, Soongsil University
★ Corresponding author: sslee@ssu.ac.kr, 02-820-0692

※ Acknowledgment
"This work was supported by the Industrial Core Technology Development Program (10049095, "Development of Fusion Power Management Platforms and Solutions for Smart Connected Devices) funded by the Ministry of Trade, Industry & Energy."
Manuscript received Dec 18, 2015; revised Dec 29, 2015; accepted Dec 30, 2015.

Ⅰ. 서론

8051 마이크로콘트롤러는 크기가 작고 명령어 집합이 간단하여 주로 SoC (system-on-chip) 상에서 전체 시스템의 간단한 통합 제어를 위해 많이 사용된다. 8051 마이크로콘트롤러는 비동기적인 데이터 전송을 위해 UART[1] (universal asynchronous receiver/transmitter)를 사용하고 있는 경우가 많은데, 최근 SoC의 발전 경향에 비추어보면 비동기적으로 제어 및 전송해야 하는 센서 및 액츄에이터 등의 수가 증가함에 따라

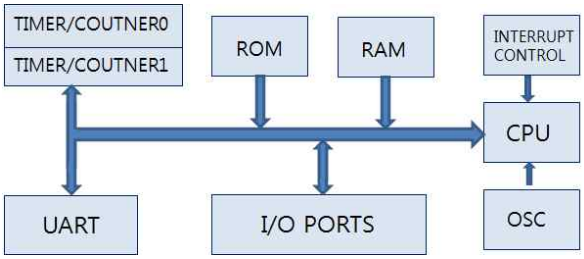


Fig. 1. Block diagram of OC8051 microcontroller
그림 1. OC8051의 블록도

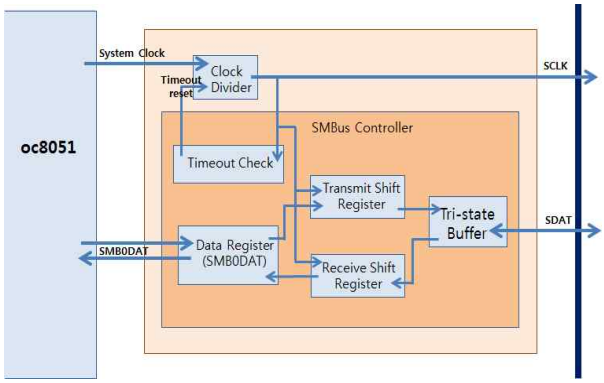


Fig. 2. Block diagram of SMBus
그림 2. SMBus의 블록도

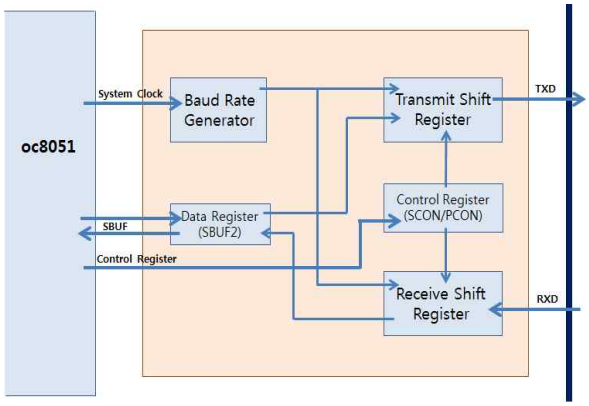


Fig. 3. Block diagram of UART
그림 3. UART의 블록도

UART 하나로는 모자라는 경우가 자주 발생한다. 또한, 시스템의 배터리 및 전력 제어 등에 많이 쓰이는 SMBus[2] (system management bus)의 필요성도 점차 늘어나고 있다.

8051 마이크로콘트롤러는 GPIO (general purpose I/O) 버스를 통해 외부 블록을 제어하고 데이터를 전송하는데, 대부분의 외부 블록은 시스템 클록에 맞춰 동기적으로 동작하기 때문에 GPIO를 사용해도 속도에 큰 문제가 없지만, UART나 SMBus와 같이 시스템 클록에 비해 매우 낮은 속도로, 그것도 비동기적으로 데이터를 전송하는 I/O 포트의 경우에는 단지 몇 바이트의 데이터를 전송하는 데에도 마이크로콘트롤러가 상당히 오랜 동안 동작을 멈추고 데이터의 전송이 끝나기를 기다려야 한다. 이러한 문제는 여러 개의 비동기 포트가 필요할 때 더욱 심각해진다.

본 논문에서는 이러한 문제점을 해결하기 위해 2개의 UART 포트와 1개의 SMBus를 탑재하고, 이들 I/O 포트를 운용하기 위한 데이터 전송 전용 명령어를 추가한 8051 마이크로콘트롤러를 설계하였다. 소스코드의 형태로 공개된 그림 1의 OC (open code) 8051[3]을 기반으로 설계하였으며, 여러 디바이스를 동시에 제어하고 데이터를 전송할 수 있다[4][5]. 또한 데이터 전송 전용 명령어의 추가를 통해 응용 프로그램의 코드 크기를 감소시킬 수 있으며, 다수의 디바이스와 데이터를 주고받는 동안에도 마이크로콘트롤러가 멈춰 있지 않고 다른 프로그램을 수행할 수 있어서 동작 효율을 크게 향상할 수 있다.

II. SMBus 및 UART의 추가

SMBus는 그림 2와 같이 구성되며 하나의 마스터 디바이스와 다수의 슬레이브 디바이스를 하나의 버스로 연결하고 다수 바이트의 데이터를 한 번에 주고받을 수 있다. 전송 속도는 10kHz에서 100kHz로 전송 속도가 9600bps인 UART보다 빠르다. UART처럼 두 개의 선으로 데이터 전송이 가능하지만 시스템 클록에 동기되어 동작한다. SMBus의 동작 주파수는 SCLK의 형태로 시스템 클록에 동기되며 외부의 슬레이브 디바이스도 SCLK에 맞춰 동작하게 된다[6].

UART는 그림 3과 같이 구성된다. 다수 디바이스와 데이터 전송이 가능한 SMBus와는 달리 UART는 오직 하나의 디바이스와만 데이터를 전송할 수 있다.

본 논문에서는 UART를 2개 탑재하여 서로 다른 GPIO로 2개의 디바이스와 각각 UART 전송을 수행할 수 있도록 하였다. 2개의 UART를 동시에 사용하는 경우 데이터 레지스터나 전송 완료 플래그를 하나로 사용하게 되면 데이터가 겹칠 수 있으므로, 제어 레지스터는 공유하되 이외의 레지스터들은 하나씩 더 추가하여 따로 사용하도록 한다. 마찬가지로 송수신할 데이터를 임

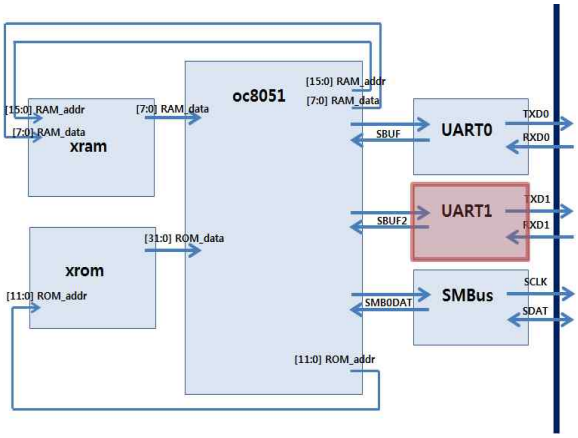


Fig. 4. Block diagram of OC8051 with additional SMBus and UART
그림 4. OC8051에 SMBus와 UART를 추가한 블록도

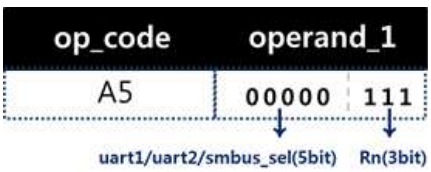


Fig. 5. Structure of the proposed application-specific transmission instruction
그림 5. 제안한 전송 전용 명령어의 구조

시로 저장할 버퍼 레지스터인 SBUF도 하나씩 더 추가하여 따로 사용한다.

그림 4는 기존 OC8051에 SMBus와 UART 포트를 추가한 모습이며, 3개의 비동기 인터페이스마다 각각의 데이터 레지스터를 갖고 서로 다른 GPIO를 통해 데이터를 전송한다.

III. 데이터 전송 전용 명령어의 추가

8051은 외부와 데이터를 주고받는 데이터 전송 명령어가 별도로 없기 때문에, UART나 SMBus와 같은 데이터 전송 포트를 사용하려면 표 1과 같이 여러 개의 명령어를 사용하여야 한다. 따라서 프로그램의 코드 크기가 증가하고 수행 시간이 길어지는 단점이 있다. 이러한 문제점을 해결하기 위해 본 논문에서는 이들 데이터 전송 포트를 위한 전용 명령어를 개발하였다.

8051의 명령어는 8비트 오피코드인 0x00 ~ 0xFF를 사용하지만, 이중에서 0xA5를 제외한 나머지는 이미 모두 사용 중이다. 따라서 본 논문에서는 비어있는 0xA5 하나를 두 개의 UART 및 SMBus의 송·수신을 위한 전송 전용 명령어

Table 1. Example of data transmission program in conventional 8051 microcontrollers
표 1. 기존 8051 마이크로콘트롤러의 데이터 전송 프로그램 예

Serial Interface		Hex Code	Assembly Code
UART	Tx	30 99 FD 8F 99 C2 99	JNB TI, tx_data MOV SBUF, R7 CLR TI
	Rx	30 98 FD C2 98	JNB RI, 02C8H CLR RI

SMBus	Tx	E5 CA
		60 FC
		8F C0
		75 CA 00
			MOV A, SMB_INT
			JZ 0507H
			MOV SMB0DAT, R7
			MOV SMB_INT, #00H

	Rx	E5 CA
		60 FC
		75 CA 00
			MOV A, SMB_INT
			JZ 0525H
			MOV SMB_INT, #00H

Table 2. Proposed application-specific transmission instructions
표 2. 제안한 데이터 전송 전용 명령어

Serial Interface		Instruction (opcode+operand)	Hex Code
UART0	Tx	10100101 **00000**xxx[a]	A5 07[b]
	Rx	10100101 **00001**xxx	A5 0F
UART1	Tx	10100101 **00010**xxx	A5 17
	Rx	10100101 **00011**xxx	A5 1F
SMBus	Tx	10100101 **00100**xxx	A5 27
	Rx	10100101 **00101**xxx	A5 2F

a) xxx denodes register number (000~111).
b) Hex code is shown when R7 is assumed.

로 사용한다.

전용 명령어는 오피코드가 0xA5로 모두 동일하기 때문에 그림 5와 같이 오퍼랜드 내에 상위 5비트로 I/O 포트의 종류를 나타내며 하위 3비트로 범용 레지스터의 종류를 나타낸다. 표 2는 새로 추가된 모든 전송 전용 명령어의 구조를 나타낸 것이다. 오퍼랜드 상위 5비트인 32개의 자리 중 6개만 사용 중이므로 추후에 다른 I/O 포트를 더 추가하기 위한 여유는 충분하다.

기존 명령어는 데이터를 전송하는 동안 PC (program Counter)가 다음 명령어를 읽지 않도록 하기 위해 반복문을 이용하며, 송·수신 완료 플래그가 1이 될 때까지 그 값을 계속 확인하면서 PC가 같은 자리를 돈다. 또한 전송이 완료된 후

Table 3. Byte counts of instructions for data transmission
표 3. 데이터 전송에 필요한 명령어 바이트 수

Conventional multiple instructions in Table 1			Proposed application-specific instruction in Table 2		
UART	Tx	7 bytes	UART	Tx	2 bytes
	Rx	5 bytes		Rx	2 bytes
SMBus	Tx	9 bytes	SMBus	Tx	2 bytes
	Rx	7 bytes		Rx	2 bytes

Table 4. Cycle counts of instructions for data transmission
표 4. 데이터 전송에 필요한 명령어 사이클 수

Conventional multiple instructions in Table 1			Proposed application-specific instruction in Table 2		
UART	Tx	5 cycles	UART	Tx	2 cycles
	Rx	3 cycles		Rx	2 cycles
SMBus	Tx	7 cycles	SMBus	Tx	2 cycles
	Rx	5 cycles		Rx	2 cycles

에는 제어코드 상에서 전송 완료 플래그를 다시 리셋해 주는 과정이 필요하기 때문에 송·수신에 표 1과 같이 여러 개의 명령어가 필요하다.

본 논문에서 제안한 전용 명령어에서는 하나의 명령어로 데이터 전송을 완료하기 위해 새로운 인터럽트를 추가했다. 즉, 오피코드가 전용 명령어 0xA5이면서 데이터가 전송 중일 때 인터럽트를 발생시키고, 인터럽트가 1인 동안에 PC가 제자리에 멈추어 다음 명령어를 읽어오지 않는 방법을 사용하였다. 전송이 완료된 후에는 하드웨어적으로 인터럽트가 다시 리셋되어 PC가 다시 동작하게 된다.

특히 앞선 내용에서도 알 수 있듯이, 기존 명령어는 송·수신 완료 플래그가 1이 되는지를 계속 확인하면서 PC가 같은 자리를 돌아야 하므로 데이터 전송 동안 마이크로콘트롤러가 다른 동작을 할 수가 없다. 하지만 제안한 전송 전용 명령어에서는 데이터 전송 동안 다른 동작도 함께 수행할 것인지를 선택하는 플래그를 추가하여 전용 명령어로 데이터를 전송하는 동안 다른 동작을 동시에 병행할 수 있다.

표 2에서 새로 제안한 데이터 전송 전용 명령어는 기존 8051 마이크로콘트롤러에서 표 1과 같이 작성되는 데이터 전송 프로그램과 동일한 기능을 가지면서도 명령어 길이가 2 바이트밖에 되지 않으므로 표 3과 같이 명령어의 바이트 수가 크게 감소한다. 이에 따라 응용 프로그램의 코드 크기를 크게 감소시킬 수 있다. 비슷한 이유로 명령어 바이트 수뿐만 아니라 명령어 사이클 수도 표 4와 같이 크게 감소한다.

Ⅳ. 구현 및 검증

먼저 제안한 8051 마이크로콘트롤러를 Verilog HDL로 기술하고 시뮬레이션을 수행하였다. 그림 6 (a), (b)는 UART의 시뮬레이션 파형을 나타낸 것이다. 데이터 전송 동안 인터럽트 신호가 1이 되고 동시에 PC가 멈춘다. 이후 1바이트 전송이 끝나면 인터럽트가 0이 되면서 PC도 다시 동작하여 다음 명령어를 읽어오는 것을 확인할 수 있다. 또한, 수신 데이터 레지스터의 값을 통해 전용 명령어를 이용한 UART0과 UART1의 통신이 제대로 이루어졌음을 알 수 있다.

그림 6 (c), (d)는 SMBus의 시뮬레이션 파형을 나타낸 것이다. 1바이트 데이터 전송마다 인터럽트가 1이 되고 인터럽트가 1인 동안 PC가 멈추며 데이터가 잘 전송되는 것을 확인할 수 있다.

그림 7은 동시 동작 플래그를 세트함으로서 하나의 포트에서 데이터 전송을 수행하면서 틈틈이 다른 동작을 수행하는 것이 가능하다는 것을 확인한 결과이다. 동시 동작 플래그가 리셋 상태이면 그림 7 (a)와 같이 UART 포트의 데이터 전송만 수행되고 나머지 시간은 마이크로콘트롤러가 멈춰있기 때문에 현재 수행되는 명령어를 나타내는 op_cur이 중간 중간 멈춰있지만, 동시 동작 플래그를 세트해주면 그림 7 (b)와 같이 UART 포트에서 데이터를 전송하는 틈틈이 다른 동작을 수행하기 때문에 op_cur의 값이 계속 변화하는 것을 볼 수 있다.

또한, 시뮬레이션이 완료된 8051 마이크로콘트롤러를 FPGA 보드에서 합성하고 동작을 검증하였다. UART의 동작은 FPGA 보드와 PC를 연결하고 SerialCom 소프트웨어를 통해 9600 bps로 ASCII 코드 형태로 데이터를 주고받도록 하였다. 검증 결과, FPGA 보드 내에서 합성된 8051 마이크로콘트롤러가 성공적으로 두 개의 UART를 통해 데이터를 주고받을 수 있음을 그림 8과 같이

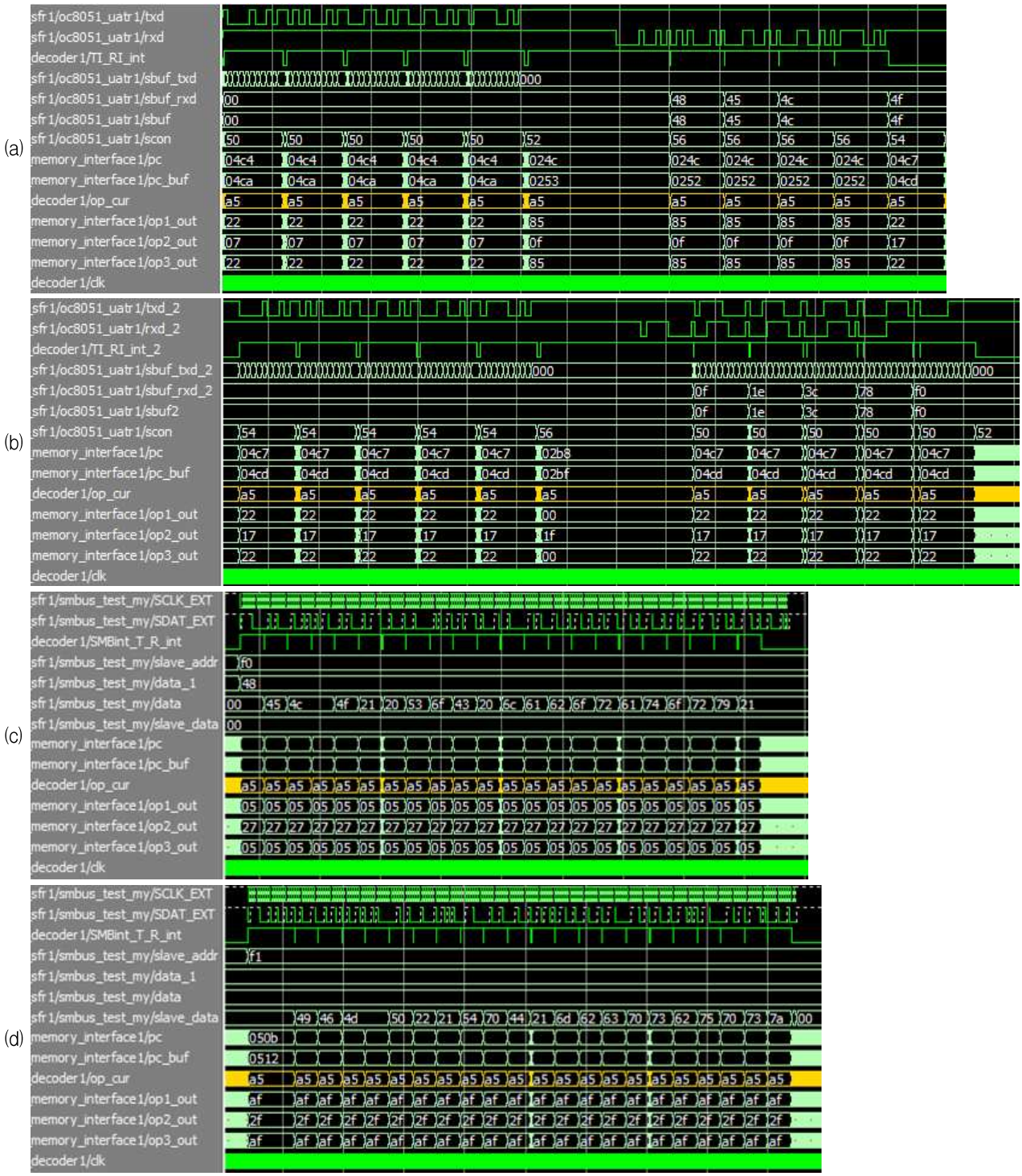


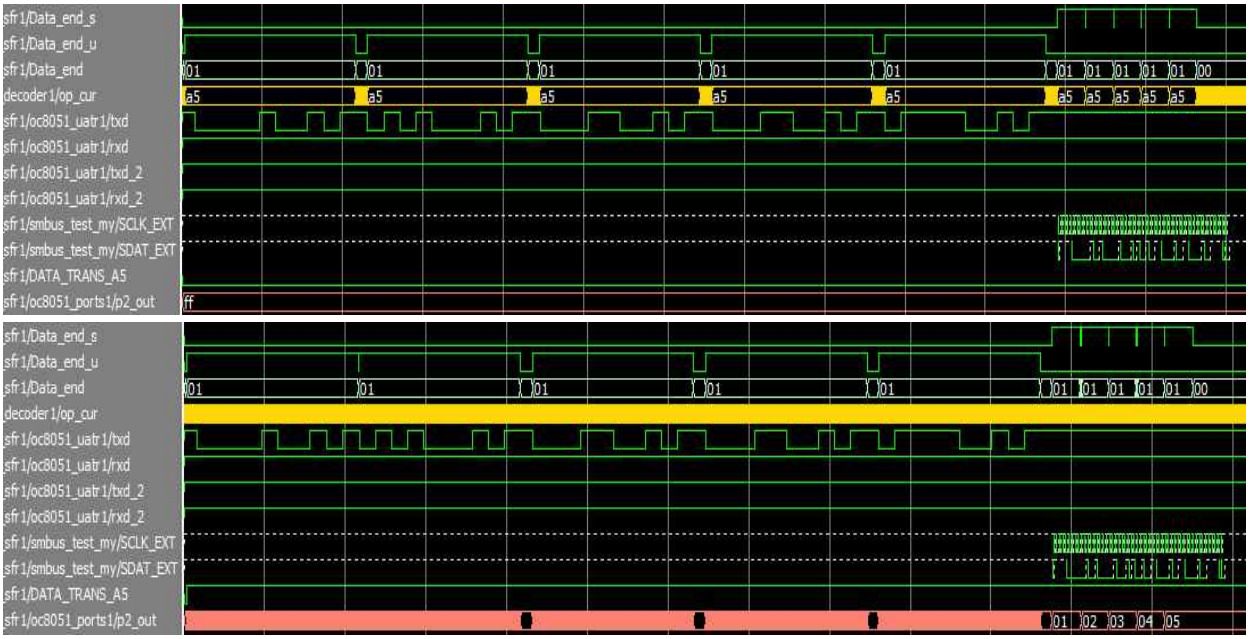
Fig. 6. Simulation waveform (a) UART0 (b) UART1 (c) SMBus Tx (d) SMBus Rx
그림 6. 시뮬레이션 파형 (a) UART0 (b) UART1 (c) SMBus Tx (d) SMBus Rx

확인하였다.

UART와 달리 SMBus는 SerialCom 소프트웨어로 통신을 확인할 수가 없다. 따라서 Xilinx ISE에서 지원되는 툴의 하나인 Chipscope를 이용하여 FPGA에서 SMBus의 동작 파형을 확인하여 검증하였다.

그림 9는 Chipscope를 이용하여 확인한 SMBus의 송신 및 수신 파형이며 시스템 클럭은 24MHz, SMBus의 동작 주파수는 100kHz이다. 두 파형에서 공통으로 나타나 있는 SCLK와

(a)

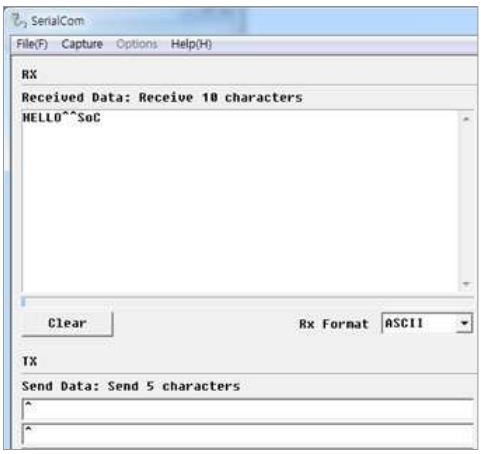


(b)

Fig. 7. Simulation waveform of concurrent operation flag (a) Reset (b) Set

그림 7. 동시 동작 플래그의 시뮬레이션 파형 (a) 리셋 (b) 세트

(a)



(b)

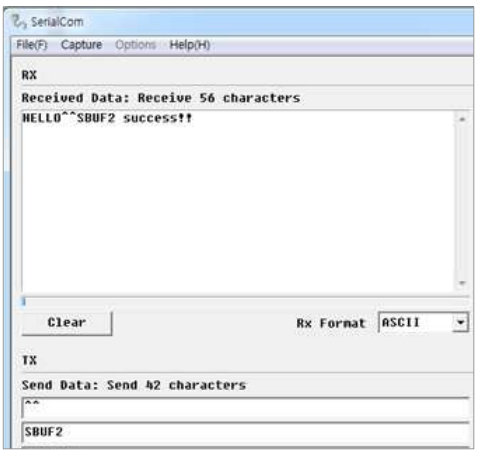
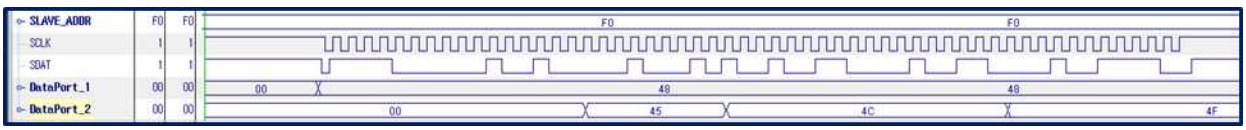


Fig. 8. UART operation of FPGA board confirmed by SerialComm (a) UART0 (b) UART1

그림 8. FPGA 보드 상에서 SerialCom으로 확인한 UART 동작 (a) UART0 (b) UART1

(a)



(b)

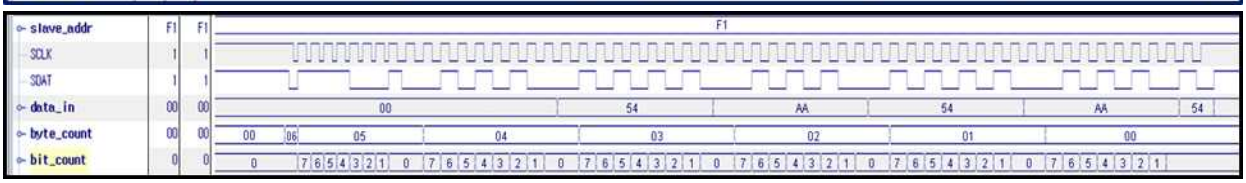


Fig. 9. SMBus waveform of FPGA board measured by Chipscope (a) Tx (b) Rx

그림 9. FPGA 보드 상에서 Chipscope로 측정된 SMBus 파형 (a) Tx (b) Rx

SDAT는 SMBus의 클록과 데이터 신호선이며, 아래 신호들은 송신과 수신 데이터 레지스터 및 비트 수를 나타낸다. 검증 결과, FPGA 보드 내에서 합성된 8051 마이크로콘트롤러가 성공적으

Table 5. Gate count
표 5. 게이트 수

Conventional 8051 microcontroller	Proposed 8051 microcontroller	Increment
29949 gates	30229 gates	0.9 %

로 SMBus를 통해 데이터를 주고받을 수 있음을 확인하였다.

전용 명령어를 추가함에 따라 게이트 수도 함께 증가하게 되는데, Design Compiler 툴을 사용하여 0.18 um 공정에서 합성한 결과는 표 5와 같으며 전송 전용 명령어의 추가 전후에 게이트 수 차이가 거의 없음을 알 수 있다.

Ⅴ. 결론

본 논문에서는 센서, 액츄에이터, 배터리 등 다양한 종류의 여러 디바이스와 연결할 수 있도록 8051 마이크로프로세서에 2개의 UART와 1개의 SMBus를 탑재하였고, 이들 직렬 인터페이스를 쉽고 빠르게 제어할 수 있도록 데이터 전송 전용 명령어를 추가하였다. 또한 데이터 전송 동안 다른 동작도 함께 수행할 것인지를 선택하는 플래그를 추가하여 전용 명령어로 데이터를 전송하는 동안 다른 동작을 동시에 병행할 수 있다. 이에 따라 응용 프로그램의 코드 크기를 감소시킬 수 있으며, 다수의 디바이스와 데이터를 주고받는 동안에도 마이크로콘트롤러가 멈춰있지 않고 다른 프로그램을 수행할 수 있어서 동작 효율을 크게 향상할 수 있다.

References

[1] J. Kim and U. Choi, “A Low Power smartRF Transceiver Hardware Design For 2.4 GHz Applications”, Journal of IKEEE, vol. 12, no. 2, pp. 75-80, Jun. 2008.

[2] C. Oh and S. Lee, “A Study on Implement of Smart Battery Management System Using Embedded Processor”, Journal of IKEEE, vol. 15, no. 4, pp. 345-353, Dec. 2011.

[3] Opencores, "OpenCore 8051", http://opencores.org/project,8051

[4] Y. Jang, “A Study of Low-Power Sensor Signal Processing SoC Platform”, Doctoral Dissertation, Soongsil University, 2010.

[5] J. Kim, “A Study on Adding Serial Interface for Multi-Byte Transmission of OC8051 Microprocessor”, to appear in Master Dissertation, Soongsil University, 2016.

[6] K. Seo, “Design and Implementation of System Management Bus Interface to Control the Smart Battery,” Doctoral Dissertation, Yeungnam University, 2006.

BIOGRAPHY

Jihye Kim (Student Member)

2014: BS degree in Electronic Engineering, Soongsil University.
2014~Now: MS candidate in Electronic Engineering, Soongsil University
<Main Interest> Microprocessor, Power Management

Seongsoo Lee (Life Member)

1991: BS degree in Electronic Engineering, Seoul National University.
1993: MS degree in Electronic Engineering, Seoul National University.
1998: PhD degree in Electrical Engineering, Seoul National University.
1998~2000: Research Associate, University of Tokyo.
2000~2002: Research Professor, Ewha Womans University.
2002~Now: Professor in School of Electronic Engineering, Soongsil University.
<Main Interest> HEVC, Low-Power SoC Design, Multimedia SoC Design, Battery Management