모토롤라 MPC8XX 마이크로프로세서와 데이터 저장장치간 고속 데이터 입/출력부 설계 및 구현

김 기 홍, 이 승 수, 황 인 호
국가보안기술연구소
전화 : 042-860-1425 / 핸드폰 : 011-9567-3040

Design and Implementation of High Speed Data I/O Block Between Motorola MPC8XX Microprocessor and Memory Devices

Ki Hong Kim, Seung Soo Lee, In Ho Hwang
National Security Research Institute
E-mail : hong0612@etri.re.kr

Abstract

In this paper, we propose a simple and efficient data input/output block with high speed between Motorola MPC8XX microprocessor and memory devices. Proposed method is capable of high speed data read and write using the address decoder and the burst cycle between Motorola PowerPC based MPC8XX microprocessor and fixed address locating memory devices such as FIFO, PCMCIA card, and so on. Experimental results are given our findings and discussions.

Ⅰ. 서론

다양한 응용 프로그램의 수용과 마이크로 프로세서의 성능향상 등의 추세에 맞추어 PowerPC 기반의 모토롤라 MPC8XX 마이크로 프로세서는 다양한 응용분야에 광범위하게 적용되고 있다. MPC8XX 마이크로 프로세서는 PowerPC 기반의 변형된 형태로, CPU는 명령 및 데이터 캐쉬와 MMU(Memory Management Unit) 등을 통합한 32비트 PowerPC로 구현된다. 또한 PowerPC RISC(Reduced Instruction Set Computer), CPM(Communication Process Unit), 메모리 및 인터럽트 컨트롤러, 통신용 RISC 프로세서 등을 내장하고 있는 SOC(System On Chip) 타입의 임베디드 프로세서이다[1,2].

일반적으로 FIFO 또는 I/O 타입의 PCMCIA(Personal Computer Memory Card International Association)[3] 카드 등과 같이 데이터 어드레스가 특정 어드레스에 고정되어 위치하는 메모리 디바이스와 MPC8XX 마이크로 프로세서간에 연속적인 데이터 읽기/쓰기가 요구되는 경우, CS(Chip Select) 신호[1,2] 하에 이들 디바이스에 데이터를 읽거나 쓴 후, 다음 데이터를 읽거나 쓰고자 할 경우 약 17 ~ 19 클럭 정도의 시간이 지난 후에야 다음 CS 신호가 발생된다. 따라서 이러한 CS 신호간 시간지연은 MPC8XX 마이크로 프로세서와 이들 디바이스간에 전체적인 데이터 읽기/쓰기 속도를 저하시켜 고속의 데이터 입/출력부 설계에는 한계가 있다[4].

본 논문에서는 이러한 CS 신호간 시간지연 문제를 해결하기 위해 MPC8XX 마이크로 프로세서와 특정 어드레스에 데이터 입/출력부가 고정되어 위치하는 FIFO 또는 PCMCIA 카드 등의 디바이스간에 어드레스 디코더와 버스트(burst) 사이클을 이용하여 CS 신호간 시간지연을 줄임으로써 고속의 데이터 읽기/쓰기가 가능한 데이터 입/출력부를 설계 및 구현하고자 한다. 본 논문의 구성은 다음과 같다. 먼저 Ⅱ장에서는 MPC8XX 마이크로 프로세서와 특정 어드레스에 고정되어 위치하는 메모리 디바이스간에 종래의 데이터 입/출력부를 설명하며 Ⅲ장에서는 어드레스 디코더와 버스트 사이클을 이용한 하드웨어 및 소프트웨어 성능개

선을 통한 고속의 데이터 읽기/쓰기가 가능한 데이터 입/출력부를 설명한다. Ⅳ장에서는 시험을 통해 설계 및 구현한 고속의 데이터 입/출력부를 검증하고 마지막으로 Ⅴ장에서 결론을 맺는다.

Ⅱ. 종래의 데이터 입/출력부

모토롤라 MPC8XX 마이크로 프로세서와 데이터 어드레스가 특정 어드레스에 고정되어 위치하는 FIFO 또는 I/O 타입 PCMCIA 카드 등의 디바이스간에 종래의 데이터 입/출력부 설계 및 구현 시 사용된 하드웨어 구성도와 MPC8XX 마이크로 프로세서와 어셈블리 소스 코드는 각각 그림 1 및 표 1과 같다.

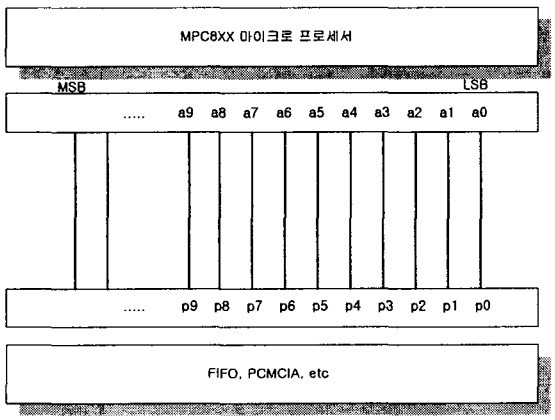

그림 1. 하드웨어 구성도

표 1. 어셈블리 소스 코드

함수	설명
_Dev_read : lhzu r0, 0(r4) sthu r0, 2(r3)	디바이스(r4)에서 데이터 읽어서 메모리(r3)로 데이터 쓰기
_Dev_write : lhzu r0, 2(r4) sthu r0, 0(r3)	메모리(r4)에서 데이터 읽어서 디바이스(r3)로 데이터 쓰기

그림 1에서 보는 것처럼, MPC8XX 마이크로 프로세서와 디바이스간 데이터 어드레스가 일대일로 맵핑되어 있으며 이는 MPC8XX 마이크로 프로세서가 지원하는 lhzu(load half word and zero with update), sthu (store half word with update) 등의 데이터 읽기 및 쓰기 기능의 어셈블리 명령어[5,6,7]를 이용하여 일정 크기 단위로 데이터 읽기/쓰기가 가능하다.

그림 2는 그림 1과 표 1의 종래의 하드웨어 및 어셈블리 소스 코드 구성을 이용하여 MPC8XX 마이크로 프로세서와 FIFO간 16비트 크기 단위로 데이터를 연속하여 읽거나 쓰는 경우, FIFO CS 신호, CS 신호간 시간지연 등의 타이밍을 측정한 것이다.

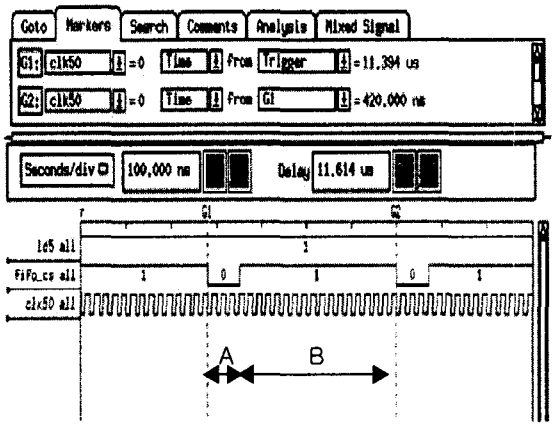

그림 2. 데이터 읽기/쓰기 타이밍도

그림 2에서 보는 것처럼, MPC8XX 마이크로 프로세서와 FIFO간에 16비트 크기 단위로 데이터를 연속적으로 읽거나 쓰는 경우 50MHz 클럭의 MPC8XX 마이크로 프로세서의 FIFO CS 시간(A)은 3 ~ 4 클럭 정도이며 다음 데이터를 읽거나 쓸 때까지의 시간 즉, CS 신호간 시간지연(B)은 18클럭 정도이다. 따라서 전체적으로 16비트 크기 단위로 데이터를 읽거나 쓰기 위해서 요구되는 시간(A+B)은 약 21 ~ 22 클럭 즉, 1/50MHz * 21(22) 클럭 = 420 ~ 440ns 정도로 이는 전체적인 데이터 읽기/쓰기 성능의 저하를 가져와 좀 더 빠른 데이터 읽기/쓰기가 요구되는 데이터 입/출력부 설계에는 사용상 한계를 가진다.

Ⅲ. 제안한 고속의 데이터 입/출력부

특정 어드레스에 고정되어 위치하는 FIFO 또는 데이터 입/출력부가 특정 어드레스에 고정되어 위치하는 PCMCIA 카드 등의 데이터 어드레스에 어드레스 디코더 회로를 추가하고 MPC8XX 마이크로 프로세서의 버스트 사이클을 이용하여 특정 어드레스를 고속으로 액세스 가능하게 한다. 본 논문에서 제안된 어드레스 디코더 회로는 액세스 하고자 하는 하나의 특정 어드레스를 MPC8XX 마이크로 프로세서 측에 대해서는 하나의 고정된 어드레스가 아닌 연속된 일련의 어드레

스로 보이도록 하여 고속으로 데이터 읽기/쓰기를 가능하게 한다.

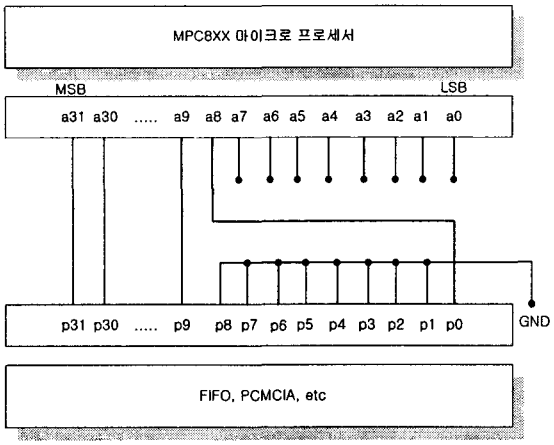

그림 3. 어드레스 디코더 구성도

그림 3은 본 본문에서 제안한 고정된 데이터 어드레스를 일정 크기의 연속된 어드레스 영역으로 확장 시 적용되는 어드레스 디코더를 나타낸 것이다. 고정된 데이터 어드레스에 위치하는 디바이스의 어드레스 영역을 디코더 회로를 추가하여 최대 1,024 비트만큼 확보하여 그 영역 내의 모든 어드레스에 데이터 읽기/쓰기가 가능하도록 구성한다. 예를 들어, FIFO의 데이터 어드레스가 0x10000000h인 경우 0x10000000h부터 0x10000100h까지의 영역을 FIFO 어드레스로 할당하여 0x10000000h에서 0x10000100h까지의 어드레스 영역은 어드레스 디코더에 의해 FIFO의 실제 데이터 어드레스인 0x10000000h 위치로 맵핑된다.

한편, MPC8XX 마이크로 프로세서가 지원하는 어셈블리 명령어 중 버스트 사이클을 지원하는 데이터 읽기 및 쓰기 기능의 명령어를 이용하여 MPC8XX 마이크로 프로세서와 특정 어드레스에 고정되어 위치하는 디바이스간에 데이터 읽기/쓰기가 가능한 데이터 입/출력부를 구성하였다.

표 2는 어셈블리 소스 코드를 나타낸 것이다. 마이크로 프로세서의 버스트 사이클을 지원하는 lmw(load multiple word), stmw(store multiple word) 어셈블리 명령어[5,6,7]를 이용하여 32비트 크기 단위로 데이터를 읽거나 쓰기가 가능하도록 구성하였다. 32비트 크기의 r8에서 r31까지의 24개의 레지스터를 이용하여 일정 크기의 데이터가 저장된 특정 어드레스에 고정되어 위치하는 디바이스(r3)로부터 lmw 명령어를 이용, 버스트 사이클로 한번에 32비트 단위로 데이터를 읽어와 r8에서 r31까지의 레지스터에 이들 데이터를 저장한 다음 이들 데이터를 stmw 명령어를 이용, 버스트 사이클로 한번에 32비트 단위로 특정 어드레스에 고정되어 위치하는 디바이스(r4)로 데이터를 전송하게 된다. 이러한 소스 코드 구성은 그림 2에서의 MPC8XX 마이크로 프로세서에 의한 디바이스 CS 신호간 시간지연을 줄이게 되어 고속의 데이터 입/출력부 설계를 가능하게 한다.

표 2. 어셈블리 소스 코드

함수	설명
lmw r8, 0(r3) lmw r9, 0(r3) .... lmw r30, 0(r3) lmw r31, 0(r3)	디바이스(r3)에서 32비트 단위로 데이터 읽기
stmw r8, 0(r4) stmw r9, 0(r4) .... stmw r30, 0(r4) stmw r31, 0(r4)	디바이스(r4)로 32비트 단위로 데이터 쓰기

Ⅳ. 설계, 구현 및 검증

본 논문에서는 그림 4에서 보는 설계 및 구현 환경을 이용하여 MPC8XX 마이크로 프로세서와 특정 어드레스에 고정되어 위치하는 디바이스간에 고속의 데이터 입/출력부를 설계 및 구현하고 이를 시험을 통해 검증하였다.

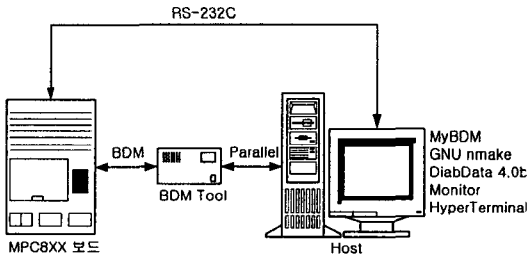

그림 4. 설계 및 구현환경

그림 4에서 보는 것처럼, BDM(Background Debugger Module) 툴은 MPC8XX 마이크로 프로세서가 장착된 프로세서 보드와 BDM 컨넥터로 연결되고 호스트 시스템과는 병렬포트로 연결된다. MPC8XX 마이크로 프로세서 보드와 호스트 시스템과는 RS-232C 컨넥터로 연결되어 캐릭터의 입/출력, 디버깅 및 모니터링 기능 등을 수행한다.

그림 5는 종래의 데이터 입/출력부 설계 시 야기되는 디바이스 CS 신호간 시간지연 문제를 해결, 고속의 데이터 읽기/쓰기가 가능한 데이터 입/출력부 설계 및 구현을 가능하게 하기 위해 본 논문에서 제안된 그림 3의 어드레스 디코더와 표 2의 어셈블리 소스 코드를 이용하여 MPC8XX 마이크로 프로세서와 고정된 데이터 어드레스를 특정 어드레스 영역으로 확장하여 위치하는 FIFO간에 데이터 읽기/쓰기 타이밍을 측정한 것이다.

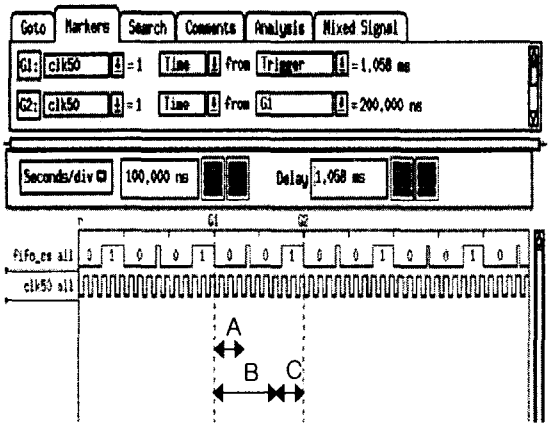

그림 5. 버스트 읽기/쓰기 타이밍도

그림에서 보는 것처럼, 데이터 어드레스가 고정되어 위치하는 FIFO의 데이터 어드레스를 일정 영역만큼 확장해서 lmw, stmw 어셈블리 명령어를 이용하여 FIFO에서 32비트 크기 단위로 일정 크기의 데이터를 연속적으로 읽어오거나 쓸 때 50MHz 클럭의 MPC8XX 마이크로 프로세서의 16비트 데이터에 대한 FIFO CS 시간(A)은 3 ~ 4 클럭 정도이고 32비트 데이터에 대한 전체적인 CS 시간(B)은 7 ~ 8 클럭 정도가 된다. 또한 다음 데이터를 읽거나 쓸 때까지의 시간 즉, CS 신호간 시간지연(C)은 3클럭 정도가 된다. 따라서 버스트 사이클로 동작함으로써 그림 2에서의 18 클럭 정도의 CS 신호간 시간지연이 3 클럭 정도로 감소하였으며 32비트 크기 단위로 데이터를 읽거나 쓰기 위해서 요구되는 시간 (B+C)은 약 10 ~ 11 클럭 즉, 1/50MHz * 10(11) 클럭 = 200 ~ 220ns 정도가 된다. 이는 MPC8XX 마이크로 프로세서와 특정 어드레스에 고정되어 위치하는 디바이스간에 일정 크기의 데이터 읽기/쓰기 처리 시, 전체적인 처리속도의 향상을 가져와 고속의 데이터 입/출력부 설계 및 구현을 가능하게 한다.

V. 결론

일반적으로 모토롤라 PowerPC 기반의 MPX8XX 마이크로 프로세서와 데이터 어드레스가 특정 어드레스에 고정되어 위치하는 디바이스간에 데이터 입/출력부 설계 시 17 ~ 19 클럭 정도의 디바이스 CS 신호간 시간지연이 발생, 데이터 읽기/쓰기 성능을 저하시킨다. 이러한 디바이스 CS 신호간 시간지연 문제를 해결, 고속의 데이터 입/출력부 설계를 가능케 하기 위해 본 논문에서는 종래의 방법을 개선, 어드레스 디코더와 버스트 사이클을 이용하여 MPC8XX 마이크로 프로세서와 특정 어드레스에 고정되어 위치하는 디바이스간에 고속의 데이터 읽기/쓰기가 가능한 데이터 입/출력부를 설계 및 구현하였다. 종래의 17 ~ 19 클럭 정도의 CS 신호간 시간지연을 3 클럭 정도로 줄임으로써 전체적인 데이터 읽기/쓰기 처리속도의 성능향상을 제공하여 고속의 데이터 입/출력부 설계를 가능하도록 하였다.

본 논문에서 제안된 방법은 모토롤라 PowerPC 기반의 마이크로 프로세서와 FIFO와 같이 데이터 어드레스가 고정되어 위치하는 디바이스 또는 호스트 시스템과의 연동을 위한 데이터 입/출력부가 특정 어드레스에 고정되어 위치하는 PCMCIA 카드 등이 장착된 다양한 마이크로 프로세서 보드에 적용, 데이터 읽기/쓰기 성능을 향상시켜 고속의 데이터 입/출력이 가능할 것으로 기대된다.

참고문헌(또는 Reference)

[1] Motorola, MPC850 PowerPUICC User's Manual, 1998.

[2] Motorola, MPC860 PowerPUICC User's Manual, 1998.

[3] Michael T. Mori, The PCMCIA Developer's Guide Second Edition, Sycard Technology, 1995.

[4] 김신수, "고속 데이터 입/출력이 가능한 임베디드 시스템", 엘지전자, 국내특허 10-2000-0085491

[5] Motorola, PowerPC Microprocessor Family : The Programming Environments for 32-Bit Microprocessor, 1997.

[6] Motorola, MPCXXX Instruction Set

[7] Motorola, PowerPC Embedded Application Binary Interface, 1995.