VHDL을 이용한 확장 인터럽트 제어기의 설계

박성수* · 박승엽*

Design of the Expanded Interrupt Controller using VHDL

Seong-su Park* · Seung-yub Park*

요 약

고성능 모터제어에 이용되는 대부분의 디지털 시그널 프로세서(DSP)에는 외부 인터럽트 입력 채널이 4개 정도 제공되고 있다. 그러나, 모터제어에서 이용될 수 있는 외부 인터럽트의 수는 이보다 훨씬 많은 수이다. 8259와 같은 상용화된 프로그램 가능한 인터럽트 제어기는 8채널의 입력을 기본으로 하고 있으며 보다 많은 인터럽트 입력을 이용하고자 할 경우 이를 종속 연결하여 채널을 확장시켜야 한다. 그리고 모터제어용 마이크로프로세서와의 인터페이스에 있어서 불편한 점이 있으므로 본 논문에서는 상용화된 인터럽트 제어기가 아닌 모터제어용 마이크로프로세서와의 인터페이스에 더욱 적합한 인터럽트 제어기의 구현과 기타의 주변장치들이 구현된 프로그램 가능한 디바이스의 소자 이용률을 높히기 위하여 모터제어에 필요한 수만큼 인터럽트 입력채널을 14채널로 확장한 인터럽트 제어기를 VHDL을 이용하여 설계하였다. 이를 위하여 간단한 인터럽트 제어기의 모델을 제안하였으며, 시뮬레이션을 통해 설계된 제어기의 유용성을 확인하였다.

ABSTRACT

Most digital signal processors provide 4 external interrupt input channels. But these are not sufficient for external interrupts of motor controls. Customized programmable interrupt controller, 8259, has 8 interrupt channels. Therefore, in the case of more external interrupt channels are needed, designers must expand by cascading the 8259. And this, 8259 device, have some inconvenience of interfacing the microprocessor in motor controls. In this paper, the expanded interrupt controller with 14 sufficient interrupt input channels for motor controls is designed using VHDL on the purpose of interfacing the microprocessor to the interrupt controller more compatibly and increasing the device utilization of FPGA/CPLD designed another peripherals. The interrupt controller model and each function blocks is proposed and illustrated. Simulation result are presented to estimate the designed interrupt controller.

키워드

VHDL, interrupt controller, interrupt mask, priority control, interrupt vector

I. 서 론

마이크로프로세서의 발전에 힘입어 모터의 구동은 더욱 고속, 고정도의 제어를 실현하고 있다. 디지털 신호처리용 마이크로프로세서(DSP)도 모터 제어에 활발히 적용되고 있다. 그러나 이러한 고속, 고정도의 DSP는 인터럽트 입력을 위한 채널이 4채널 정도밖에 갖추고 있지 않으므로[1], 모터 제어에 필요

*창원대학교 전자공학과

접수일자 : 2003. 4. 3

한 주변장치들의 모든 상태에 따른 신호들을 인터럽트 신호로 적시에 입력하여 처리하기에는 불편한 면이 있다. 이러한 단점을 해결하기 위하여 8259와 같은 상용화된 프로그램 가능한 인터럽트 제어기의 적용을 고려할 수 있으나, 이 경우에도 8채널의 입력을 기본으로 하고 있으며[2], 보다 많은 인터럽트 입력을 이용하고자 할 경우 이를 종속 연결하여 채널을 확장시켜야 하므로 전체 시스템의 집적도가 떨어지고 FPGA/CPLD와 같은 디바이스의 소자 이용률 또한 개선되지 않으며, 모터 제어 기술자들이 8259를 사용함에 있어서 다음과 같은 두가지의 불편한 점이 있다.

첫째. 인터럽트 서비스 루틴(ISR) 주소의 획득 방법

초기화 과정에서 미리 인터럽트 제어기에 저장하여 둔 주소값을 읽어 들인다. 이렇게 하기 위해서는 인터럽트 서비스 루틴을 미리 정해 둔 특정한 주소에서 시작하여야 하므로 프로그램 메모리의 사용에 있어 자유도가 떨어지며 인터럽트 서비스 루틴의 위치에 구속을 받게 된다.

둘째. 인터럽트 시퀀스에 의한 지연

MCS-80/85 모드에서는 인터럽트 신호가 발생된후 3개의 INTA신호(Call code, Lower 8-bit, Higher 8-bit)에 의하여 ISR로의 완전한 분기주소를 얻을 수 있다. 8086 모드에서는 2개의 INTA신호(Dummy INTA, 8-bit pointer)에 의하여 ISR로의 완전한 분기주소를 얻는다. 이러한 시퀀스는 인터럽트가 요구되고 난 이후 ISR로의 완전한 분기주소를 얻을 때까지 약 1590ns의 지연을 가진다. 이러한 시퀀스는 모터 제어 기술자로서는 인터럽트 수신에 너무 많은 시간을 소요하는 것이 된다. 그림 1에 8259의 인터럽트 시퀀스를 보였다[2].

본 논문에서는 위의 2가지 문제점을 다음과 같이 해결하고자 하였다.

첫째. 인터럽트 서비스 루틴(ISR) 주소의 획득 방법

미리 인터럽트 제어기에 주소값을 저장하지 않고, 인터럽트 발생시 인터럽트 번호에 대한 값만을 얻어 상대적으로 분기할 수 있도록 한다.

둘째. 인터럽트 시퀀스의 단축

인터럽트 신호가 입력되면 한번의 읽기 동작으로 인터럽트 번호에 대한 값을 얻도록 한다.

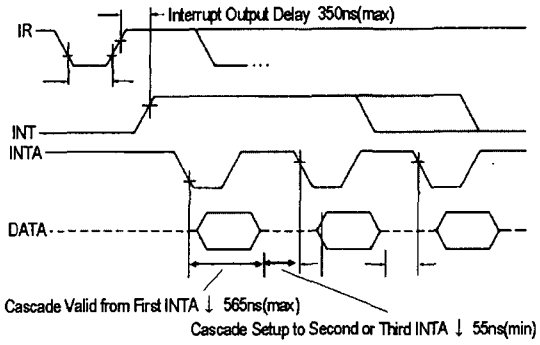


그림 1. 8259의 인터럽트 시퀀스
Fig. 1 Interrupt sequence of 8259

이와 같은 기능을 가지는 인터럽트 제어기를 구현함에 있어서 본 논문에서는 모터제어에 필요한 주변장치들을 구현한 프로그램 가능한 디바이스의 소자 이용률을 높힘과 동시에 주변장치들을 더욱 집적화할 수 있도록 하기 위하여 모터제어에 필요한 수만큼 인터럽트 입력채널을 확장한 인터럽트 제어기를 VHDL을 이용하여 설계하였다. II장에서 모터제어를 위하여 이용 가능한 인터럽트의 종류와 인터럽트 제어기가 갖춰야 할 기본 기능을 분석하였으며, III장에서 설계모델을 제안하고, VHDL을 이용하여 이들 기능 블럭을 설계하였으며 회로모듈과 동작 흐름도로서 설명하였다. IV장에서 MAX+Plus II에서 ALTERA사의 FPGA인 ACEX1K 시리즈인 EP1K100Q208-3을 목적 디바이스로 한 시뮬레이션을 통하여 설계된 인터럽트 제어기의 유용성을 검증하였다.

II. 인터럽트 제어기의 특성

모터 제어를 함에 있어서 필요한 각 주변장치로부터의 인터럽트 신호를 분석하여 보면 아래와 같다. 본 논문에서는 3상 대칭 PWM제어와 3 채널 이상을 갖춘 A/D변환기, 속도엔코더 및 호스트 PC와의 통

신을 위한 별도의 직렬통신장치를 적용한 하드웨어를 기준으로 하였다.

■ 인터럽트 처리 이벤트 수 : 14
. 3상 반도체 전력스위치로부터의 과전압 신호 3
. 3상 반도체 전력스위치로부터의 과전류 신호 3
. 3상 반도체 전력스위치로부터의 과열 신호 3
. 대칭 PWM 신호발생기의 카운터 언더플로우 신호 1
. 대칭 PWM 신호발생기의 카운터 오버플로우 신호 1
. A/D변환기로부터의 변환종료 신호 1
. 속도 엔코더신호 처리종료 신호 1
. 외부 직렬통신장치로부터의 수신완료 신호 1
■ 인터럽트 처리시간 : 500nsec(max)
■ 기본 기능
. 인터럽트 신호의 에지를 선택 기능
. 노이즈 신호 제거 기능
. 인터럽트의 우선순위 제어기능
. 개별 인터럽트 제어 레지스터의 읽기/쓰기가 가능
. 전체 인터럽트 제어 레지스터의 읽기/쓰기가 가능
. 래치된 인터럽트의 읽기/쓰기가 가능
. 제어 레지스터 및 인터럽트 벡터의 읽기/쓰기를 위한 주소 및 데이터 버스를 가짐

III. VHDL에 의한 인터럽트 제어기의 설계

1. 인터럽트 제어기의 구성

인터럽트 제어기는 다음과 같은 6개의 모듈로 구성하였다.

(1) 인터럽트 입력부
(2) 에지 선택 및 저역 통과 필터부
(3) 인터럽트 허가 및 금지부
(4) 인터럽트 래치부
(5) 우선순위 제어 및 벡터주소 생성부
(6) 인터럽트 요구신호 출력 및 벡터주소 출력부

각 모듈에 대한 기능과 입, 출력 핀에 대한 기본적인 기능은 다음과 같다.

먼저 인터럽트 입력부는 입력 전용으로 외부의 각 디바이스들로부터의 인터럽트신호를 입력한다. 인터럽트 신호를 받아들이는 입력 방법을 두가지로 대별한다면, 일정한 전압레벨이상의 신호가 의도된 시간동안 유지되면 하나의 인터럽트 신호로 인정하는 레벨트리거 방법과, 상승 또는 하강시의 에지가 발생한 후 의도된 시간만큼의 전압레벨이 유지되면 하나의 인터럽트 신호로 인정하는 에지트리거 방법으로 나눌 수 있다. 레벨트리거 방법은, 인터럽트 신호가 유효한 전압레벨을 의도한 시간이상 유지하고 있는 경우 동일한 인터럽트가 연속적으로 발생되게 되므로 외부의 인터럽트 신호 출력부에서는 일정한 전압레벨과 유지시간을 동시에 관리하여야 한다. 반면에, 에지트리거 방법은 입력신호가 일정한 전압레벨을 유지하고 있더라도 상승 또는 하강의 유효한 에지가 재발생 되지 않는 한 인터럽트는 연속적으로 발생되지 않아야 한다.

에지 선택 및 저역 통과 필터부는 외부장치의 출력신호 형태에 따라 인터럽트 입력시 상승 또는 하강 에지를 선택할 수 있어야 하며, 저역 통과 필터부는 의도한 펄스폭 이하인 경우에는 노이즈로 인식하여 유효한 입력으로 취급하지 않도록 하는 기능을 가져야 한다. 노이즈로 인식하기 위한 펄스폭은 마이크로프로세서에서 처리 가능한 수준과 외부 장치의 출력형태에 따라 다를 수 있다.

인터럽트의 허가 및 금지부는 개별 인터럽트의 허가 및 금지를 설정할 수 있도록 읽기/쓰기가 가능한 개별 마스크 레지스터와 이들 허가된 인터럽트 신호들 중 우선순위에 의하여 최종적으로 마이크로프로세서에 요구되는 인터럽트를 최종적으로 허가 및 금지할 수 있는 읽기/쓰기 가능한 글로벌 마스크 플래그로 구성된다. 글로벌 마스크 레지스터는 마이크로프로세서로 한번의 인터럽트가 요구되면 즉시 자동으로 금지되어 마이크로프로세서가 요구된 인터럽트를 인식하기 전에는 다른 인터럽트의 요구가 발생되지 않도록 한다.

인터럽트 래치부는 인터럽트 마스크 레지스터에 의하여 필터링된 입력신호들이 발생된 순위에 따라 마이크로프로세서에의 인터럽트 요구를 위하여 대기하는 동안 저장되어 개별 인터럽트들의 발생여부를 기억하는 기능을 가지며, 의도하지 않은 시간에 이미

발생되어 인터럽트 제어기에 받아들여진 경우에는 이를 선택적으로 취소할 수 있도록 하는 기능을 가진다. 그러기 위해서는 마이크로프로세서에서 읽어 확인한 후, 원하는 인터럽트 플래그만을 취소할 수 있도록 인터럽트 플래그 레지스터의 값을 조작, 저장할 수 있어야 하므로 읽기/쓰기가 가능한 레지스터로 구성되어야 한다. 이러한 기능은 인터럽트가 허가되어 인터럽트 마스크 레지스터를 통과한 입력신호라 할지라도 상황에 따라 이를 취소하여야 할 경우를 만족시키기 위한 것이다.

다음으로 우선순위 제어 및 벡터주소 생성부는, 동시 입력된 인터럽트 신호를 우선순위에 따라 차례대로 인터럽트 요구신호를 발생시킴과 동시에, 요구된 인터럽트에 해당하는 인터럽트 서비스 루틴으로의 분기제어를 위한 주소값을 생성시킨다. 우선순위는 먼저 입력의 발생순위를 최우선으로 하고 인터럽트가 동시에 2개 이상 입력된 경우는 이미 결정된 우선순위에 따라 재배열하여 마이크로프로세서로의 인터럽트 요구신호 출력시기를 제어하게 된다. 인터럽트 INT0와 INT13이 동시에 입력된 경우 우선순위가 높은 순서대로 인터럽트 요구신호를 발생시키는 개념을 그림 2에 나타내었다.

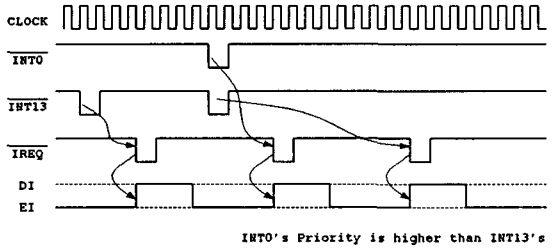


그림 2. 동시 입력된 인터럽트의 우선순위 제어
Fig. 2 Priority control of concurrent interrupts

마지막으로 인터럽트 요구신호 출력 및 벡터주소 출력 핀은 상기 인터럽트 마스크 레지스터 중 글로벌 인터럽트 마스크 레지스터에 인터럽트가 허가되어 있는 경우, 마이크로세서의 인터럽트 입력 핀으로 마이크로세서에서 요구되는 펄스폭과 전압레벨을 가지는 펄스성 신호를 출력하며, 마이크로프로세서로부터의 인터럽트를 수신하였음을 알리는 신호가 있을 때 마이크로프로세서의 읽기 동작 특성에 맞도록 벡터주소를 출력한다. 또한, 인터럽트 요구신호 출력시에는 글로벌 마스크 플래그를 금지상태로 함으로써, 마이크로프로세서에서 인터럽트 요구를 완전하게 수신하기 이전에 또다른 인터럽트 요구신호가 발생되는 것을 방지할 수 있어야 하고, 마이크로프로세서에서는 인터럽트 요구신호를 받은 후 또다른 인터럽트를 수신하기 위하여 인터럽트 마스크레지스터에 접근하여 글로벌 인터럽트 마스크를 허가상태로 재설정할 수 있어야 한다. 마이크로프로세서에서의 벡터주소 인식은 인터럽트제어기의 벡터주소 생성부 내에 있는 인터럽트 벡터주소 레지스터의 내용을 읽어들임으로써 알 수 있게 되며, 이 값에 따라 요구된 인터럽트의 서비스루틴으로의 분기가 가능하게 된다.

2. 인터럽트 제어기의 모델

설계하고자 하는 인터럽트 제어기는 입력신호 처리부, 인터럽트 선택부, 인터럽트 신호 래치부와 우선순위 제어 및 벡터주소 생성부 등을 기본 구조로 하며 마이크로프로세서로부터 읽기/쓰기를 위하여 접근 가능한 제어 논리을 갖춘 구조로 하였다. 그림 3에 설계하고자 하는 인터럽트 제어기 모델을 나타내었다.

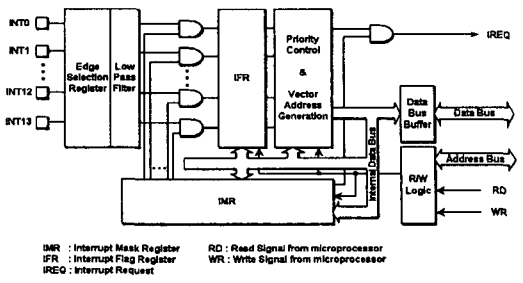


그림 3. 제안된 인터럽트 제어기 모델
Fig. 3 Proposed interrupt controller model

3. VHDL 프로그래밍

인터럽트 제어기의 설계에 있어서 개발언어로는 VHDL을 사용하였고, 개발 툴은 MAX+Plus II를 이용하였으며, FPGA는 ALTERA사의 저가형 모델인 ACEX 1K 시리즈 중 10만개의 게이트를 가진 EP1K100-3을 적용대상으로 하였다.

각 기능별 컴포넌트들은 세부적인 소자의 설계를 필요로 하지 않고 입출력관계의 알고리즘만으로 설계할 수 있는 동작적 모델링 기법을 주로 사용하였고[4], 설계한 각 컴포넌트들을 구조적 모델링으로 전체적인 회로를 구현하였다.

(1) 에지선택 레지스터(IER) 및 저역 통과 필터(LPF)

인터럽트 입력부는 14비트의 입력전용 핀으로 설정하였으며, 일반적인 산업용 마이크로컨트롤러 등에서 많이 채용되고 있는 에지트리거 방법을 채택하였다. 그리고 에지의 선택은 인터럽트 에지 선택 레지스터의 값을 '1' 또는 '0'으로 설정함으로써 가능하도록 하였다. 에지 선택 레지스터의 구성을 그림 3에 나타내었다. 에지 선택에 따른 입력신호의 판단은 에지 선택 레지스터(IER)로부터 에지 선택 정보를 입력받아, 상승 에지를 선택한 경우는 입력신호를 역전시킨 후 필터회로로 출력시키는 것으로 하였다.

본 논문에서의 저역 통과 필터(Low Pass Filter)는 FPGA의 동작주파수에 맞추어 FPGA 동작주파수의 1클럭 이하의 폭을 가지는 신호는 노이즈로 규정하였고, 클럭에 동기하여 연속 2회 동일 전압레벨을 유지한 경우에 유효한 입력으로 결정함으로써 LPF를 구현하였다. 리세트시의 에지 선택값은 일반적인 디지털 회로에서 하강 에지를 유효한 에지로 많이 적용하고 있으므로 하강 에지를 기본값으로 하였다. 그림 4는 인터럽트 입력부의 구성이다.

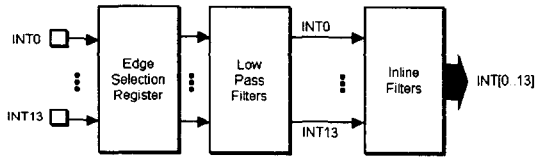


그림 4. 인터럽트 입력부
Fig. 4 Interrupt input blocks

에지 선택회로를 통과한 신호는 저역 통과 필터에 입력되어 유효한 펄스폭을 가진 신호인지의 여부를 판단하게 되고, 일단 한번의 에지에 의하여 유효한 신호로 판단되면 이후 선택된 일정한 입력 레벨을 연속적으로 유지하는 경우 인터럽트신호로서의 효력을 갖지 않도록 하므로써 레벨트리거 방식이 되지 않도록 하였다.

저역 통과 필터를 통과한 각 채널의 신호들은 다시 인라인 필터에 의하여 14비트의 병렬신호로 재구성하여 다음 단계로 보내므로써 이후의 기능 블록에서는 14비트 단위로 데이터를 취급할 수 있도록 하였다. 그림 5에 에지 선택 레지스터의 구성을 나타내었다.

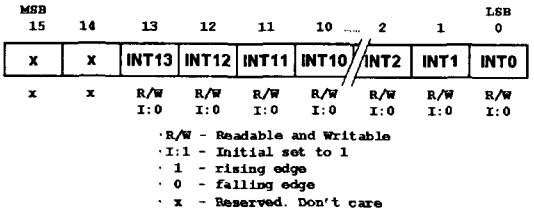


그림 5. 인터럽트 에지 선택 레지스터 (IER)
Fig. 5 Interrupt edge selection register (IER)

(2) 인터럽트 마스크 레지스터(IMR)

인터럽트 마스크 레지스터는 개별 인터럽트의 허가 및 금지를 위한 읽기/쓰기 가능한 14비트의 레지스터와 글로벌 허가 및 금지를 위한 읽기/쓰기 가능한 1비트의 글로벌 인터럽트 마스크 레지스터를 하나의 16비트 레지스터로 구성하였다.

이 레지스터는 하나의 주소를 할당하여 읽기/쓰기 신호로 접근할 수 있도록 하였고, 리세트시 기본 설정값은 모든 인터럽트를 금지 상태로 하였다. 노이즈 필터를 통과한 입력신호는 인터럽트 마스크 레지스터의 해당 비트와 논리곱을 수행하므로써 인터럽트 신호로서 받아들여질지의 여부를 결정하였다. 그림 6에 인터럽트 마스크 레지스터의 구성을 나타내었다.

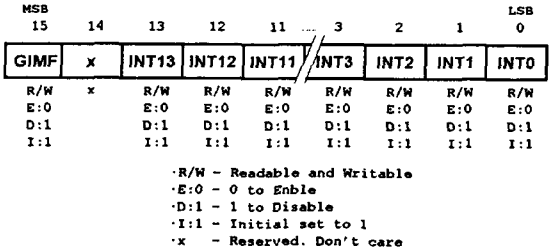


그림 6. 인터럽트 마스크 레지스터 (IMR)
Fig. 6 Interrupt mask register (IMR)

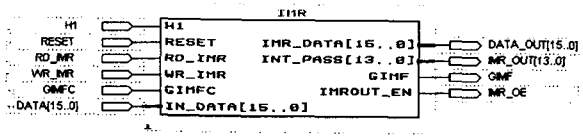


그림 7. 인터럽트 마스크 레지스터 블록
Fig. 7 Interrupt mask register block

(3) 인터럽트 플래그 레지스터(IFR)

인터럽트 래치 기능을 가지는 인터럽트 플래그 레지스터는 14비트로 구성하고 레지스터의 주소를 할당하여 마이크로프로세서에서 읽기/쓰기 신호로 접근할 수 있도록 하였다. 마이크로프로세서에서 이 레지스터를 읽어보고 원하지 않는 시점의 인터럽트가 발생되어 래치되어 있는 경우에는 그 인터럽트 플래그를 삭제할 수 있도록 하였으며, 이러한 경우는 해당되는 인터럽트의 벡터주소 및 우선순위 데이터를 동시에 삭제하여 다음 시점의 인터럽트가 유효하도록 하였다. 그리고 이미 발생하여 래치되어 있는 인터럽트와 동일한 인터럽트가 다시 발생된 경우는 무시하므로써 시간적으로 늦게 발생된 동일 인터럽트는 동시에 발생되지 않도록 하였다. 그림 8에 설계된 인터럽트 플래그 레지스터 블럭을 나타내었고, 그림 9에 인터럽트 플래그 레지스터의 구성을 나타내었다.

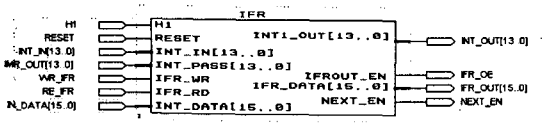


그림 8. 인터럽트 플래그 레지스터 블럭
Fig. 8 Interrupt flag register block

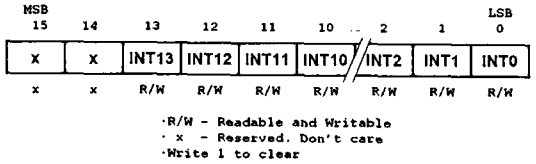


그림 9. 인터럽트 플래그 레지스터 (IFR)
Fig. 9 Interrupt flag register (IFR)

(4) 우선순위 제어 및 인터럽트 요구신호 출력부

우선순위 제어부는 허가된 인터럽트가 입력된 순서대로 적치되는 FIFO(first in first out)기능을 가진 14비트 14열 레지스터 블록으로 구성하였다. 인터럽트 플래그 레지스터에 저장되어 있는 인터럽트 신호들 중 가장 먼저 입력된 인터럽트 신호를 가진 14비트의 데이터에 의하여 마이크로프로세서로 인터럽트 요구신호를 출력하며, 만약 동시에 여러 개의 인터럽트가 입력되었다면 가장 우선순위가 높은 신호부터 인터럽트 요구신호를 발생시켜 동시에 입력된 인터럽트 신호들을 모두 인터럽트 요구신호로 처리한 후, 다음 시간에 입력된 14비트의 인터럽트 데이터를 처리하게 된다. 일단 인터럽트 요구신호를 발생하게 되면, 인터럽트 마스크 레지스터의 글로벌 인터럽트 마스크 플래그(GIMF)를 금지상태로 전환시키는 신호를 출력하여 연속적인 인터럽트의 요구신호가 발생되지 않도록 하였다. 마이크로프로세서에서는 인터럽트 서비스 루틴 내에서 인터럽트 마스크 레지스터의 글로벌 인터럽트 마스크 플래그(GIMF)를 허가상태로 전환시킴으로써 이후의 인터럽트 요구신호를 받을 수 있다. 그림 10에 설계된 우선순위 제어기 및 인터럽트 요구신호 출력부의 회로 모듈을 나타내었다. 그림 11은 우선순위 제어 및 인터럽트 요구신호 출력 알고리즘에 대한 동작 흐름도이다.

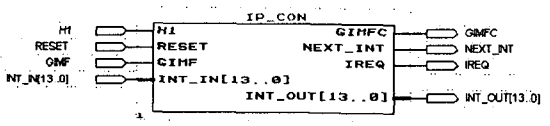


그림 10. 우선순위 제어기
Fig. 10 Priority controller

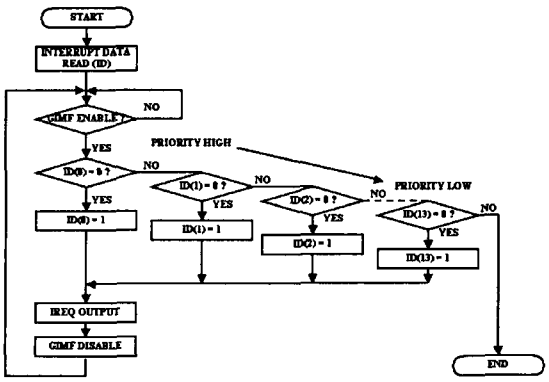


그림 11. 동시 입력된 입터럽트의 우선순위 제어 신호 흐름도
Fig. 11 Priority control flowchart of concurrent interrupts

(5) 벡터주소 생성 및 벡터주소 출력부

벡터주소의 생성은 우선순위 제어부로부터 현재 출력된 입터럽트 요구신호의 인터럽트 종류에 대한 데이터를 입력받아 벡터주소를 생성하여 벡터주소 레지스터에 보관하며, 마이크로프로세서로부터 읽기 동작이 수행되면 벡터주소 레지스터를 출력하도록 하였다. 만일 마이크로프로세서에서 현재의 인터럽트 벡터주소를 읽지 않고 글로벌 인터럽트 마스크 플래그를 허가상태로 재설정한 경우는, 현재의 주소는 소멸되고 다음에 발생하는 인터럽트의 벡터주소가 생성되게 함으로써 하나의 인터럽트만을 사용할 경우에는 필요 없는 인터럽트 벡터주소의 읽기 동작을 수행하지 않아도 되도록 하였다. 그림 12에 설계된 벡터주소 생성 및 벡터 추소 출력부의 회로 모듈을 나타내었고, 벡터주소의 출력과 마이크로프로세서의 간단한 동작 흐름도를 그림 13에, 그리고 표 1에 인터럽트 벡터 레지스터(interrupt vector register, IVR)에서 출력되는 인터럽트 벡터 주소값을 나타내었다.

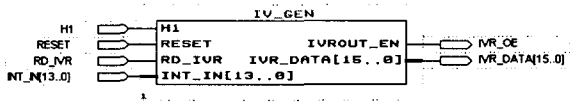


그림 12. 인터럽트 벡터주소 생성기
Fig. 12. Interrupt vector address generator

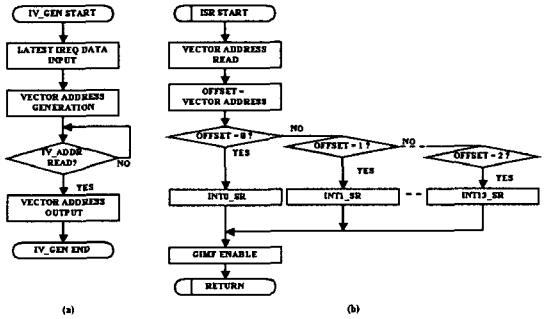


그림 13. 인터럽트 벡터주소 출력(a)과 인터럽트 서비스(b) 루틴들의 신호흐름도
Fig. 13 Flowcharts of interrupt vector address output (a) and interrupt service(b) routines

표 1. 인터럽트 벡터 레지스터의 벡터 주소값
Table 1. Interrupt vector address values of IVR

우선순위	종류	주소값
1	INT0	0000h
2	INT1	0002h
3	INT2	0004h
4	INT3	0006h
5	INT4	0008h
6	INT5	000Ah
7	INT6	000Ch
8	INT7	000Eh
9	INT8	0010h
10	INT9	0012h
11	INT10	0014h
12	INT11	0016h
13	INT12	0018h
14	INT13	001Ah

그림 14는 설계된 전체 회로의 블록도를 나타낸 것이다.

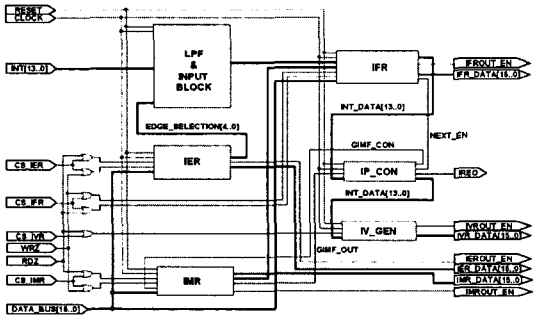


그림 14. 설계된 전체 회로도
Fig. 14 Total circuits block diagram

IV. 시뮬레이션 결과

설계된 인터럽트 제어기를 ALTERA사의 FPGA인 EP1K100QC208-3를 목적 디바이스로 하여 시뮬레이션 하였다. 입력 및 출력 신호명을 정의하면 표 2와 같으며, 표 3의 조건으로 시뮬레이션 한 결과, 그림 15에서 보인 바와 같이 글로벌 인터럽트 마스크 플래그(GIMF)가 허가 되기 전에는 입력된 어떠한 인터럽트도 마이크로프로세서로의 인터럽트 요구신호로 출력되지 않았으며, GIMF가 허가상태로 된 후, 동시 입력된 INT4와 INT6중에서 우선순위가 높은 INT4가 먼저 인터럽트 요구신호(IREQ)로 출력되었음을 알 수 있고, 이후 입력된 순서에 따라 IREQ를 각각 출력하고 또한 각각에 상당하는 인터럽트 벡터 주소값인 8(INT4, 12(INT6) 및 0(INT0)을 출력하고 있음을 볼 수 있다. 인터럽트가 입력되어 IREQ가 출력될 때까지 3클럭 사이클에 의하여 처리되었으며, 지연시간은 157ns가 소요되었다.

표 2. 시뮬레이션 신호명의 정의
Table 2. Definition of simulation signals

신호명	정 의
RESET	디바이스 리세트 신호
h1	클럭 입력
INT[13..0]	14채널 인터럽트 신호 입력 포트
WRZ	각 제어 레지스터들에의 쓰기 신호
IN_DATA[15..0]	각 레지스터에의 입력 데이터버스
CS_IER	IER에의 접근 신호
CS_IMR	IMR에의 접근 신호
CS_IFR	IFR에의 접근 신호
CS_IVR	IVR에의 접근 신호
IEROUT_EN	IER값 출력 허가신호
IMROUT_EN	IMR값 출력 허가신호
IFROUT_EN	IFR값 출력 허가신호
IVROUT_EN	IVR값 출력 허가신호
IER_DATA[15..0]	IER값 출력 데이터버스
IMR_DATA[15..0]	IMR값 출력 데이터버스
IFR_DATA[15..0]	IFR값 출력 데이터버스
IVR_DATA[15..0]	IVR값 출력 데이터버스

위의 신호에서 데이터버스를 제외한 모든 신호는 LOW Enable'이다.

표 3. 시뮬레이션 조건
Table 3. Simulation conditions

시점	조 건
h1	60ns
100ns	리세트 해제
150ns	인터럽트가 허가되기전 인터럽트 입력
250ns	INT0~INT6 허가. GIMF 금지
300ns	INT4와 INT6 동시 입력
450ns	INT0 입력
550ns	INT0~INT6 허가 및 GIMF 허가

이로서 II장에서 제시하였던 모터제어용 인터럽트 제어기의 설계기준에 부합하는 기능이 완전하게 설계되었음을 알 수 있으며, ALTERA사의 저가형 FPGA인 EP1K100QC208-3에 인터럽트 제어기를 구현함에 있어 소요된 리소스 및 성능은 표 4와 같다.

표 4. 인터럽트 제어기의 디바이스 이용률 및 성능
Table 4. Device utilization and performances

Utilization		Performance (fmax)
Logic Cells	Momory Bits	16MHz
1316 (26%)	0 (0%)	

V. 결 론

본 논문에서는 DSP 등의 고속, 고성능의 마이크로프로세서를 모터제어용으로 사용함에 있어서, 상용화된 범용 인터럽트 제어기를 사용함에 따른 문제점을 개선하기 위하여 모터 제어 전용의 인터럽트 제어기를 VHDL을 이용하여 구현하고 시뮬레이션한 결과, 다음의 결론을 얻었다.

1. 인터럽트 서비스 루틴(ISR) 주소의 획득 방법에 있어서, 기존의 8259와 같은 상용 인터럽트 제어기에서와 같이 인터럽트 제어기에 인터럽트 서비스 루틴의 주소값을 직접 저장하지 않고, 인터럽트 발생시 인터럽트 번호에 대한 값만을 얻어 이 값을 이용하여 상대적으로 분기할 수 있도록 하므로써 제어 프로그램 작성의 자유도를 높일 수 있다. 이것은 퍼스널 컴퓨터와 같이 실행될 프로그램이 메모리에 로드될 주소를 운영체제에 의하여 알 수 있는 경우가

아닌 마이크로컨트롤러 등에서는 인터럽트 서비스 루틴의 절대주소를 인터럽트 컨트롤러에 미리 지정해 두는 것은 매우 난이한 일이 되므로 제어프로그램의 작성에 있어서의 자유도에 큰 차이가 있다고 할 수 있다.

2. 인터럽트 시퀀스를 단축하기 위하여 인터럽트 신호가 입력되면 한번의 읽기 동작으로 인터럽트 번호에 대한 값을 얻도록 하므로써 상용 인터럽트 제어기에 비하여 모터 제어에 더욱 적합한 시퀀스의 구현이 가능하다.

이것은 모터제어에 있어서 반도체 전력 스위치의 단락시 또는 과전류시에 견디는 시간(withstanding time)이 10 μsec 이내이며, 이에 대한 에러신호를 전달하는 소자의 지연시간등을 감안하면(특히 광디바이스를 이용하여 제어측과 전력측을 절연한 경우, 광디바이스의 스위칭 지연시간은 수 μsec 에 달한다), 1 μsec 정도의 여유시간을 획득하는 것은 전력소자의 보호에 매우 중요한 시간이 되므로 매우 의미가 있다고 할 수 있다.

나아가서 일반적으로 VHDL로 설계하여 적용하고 있는 모터제어에 필요한 PWM신호 발생기, 엔코더 신호 처리기를 비롯한 어드레서 디코더, 프로세서 슈퍼바이저 등의 주변회로를 본 논문에서 설계된 인터럽트 제어기와 통합하여 이들 주변장치들을 구현한 디바이스의 소자 이용률의 개선뿐만 아니라 모터 제어를 더욱 용이하게 할 수 있는 향상된 주변장치들의 기능개발이 이루어져야 할 것이다.

참고문헌

[1] Texas Instruments, TMS320VC33 Digital Signal Processor Data Sheet.

[2] Intel, 8259A Programmable Interrupt Controller Data Sheet.

[3] J. A. Peter, "The designer's guide to VHDL", chapter 1, Morgan Kaufmann Publishers, San Francisco, California.

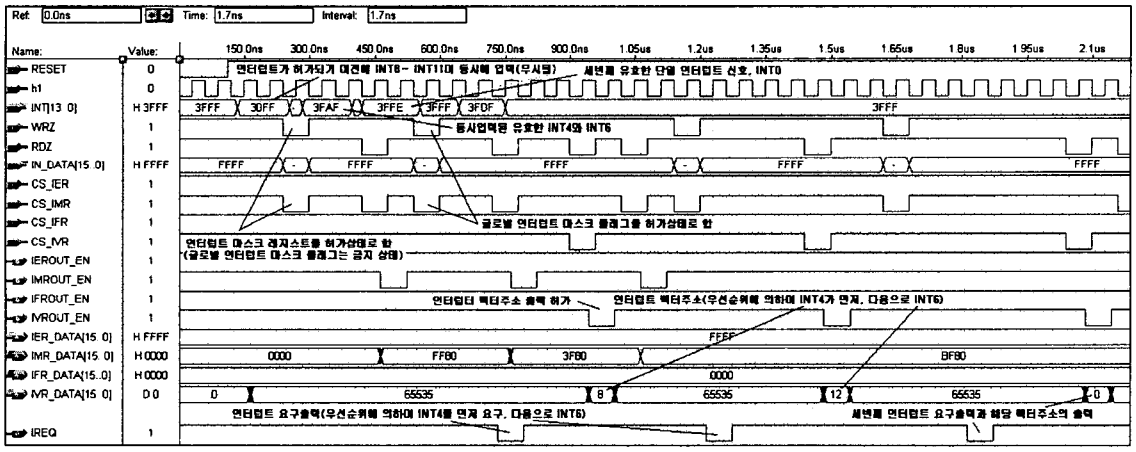
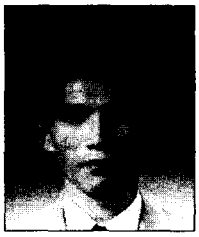


그림 15. 시뮬레이션 결과

Fig. 15 Simulation results

저자소개



박성수(Seong-Su Park)

1987년 영남대 전자공학과 졸업.
1999년 창원대 전기전자제어 공학과 석사 졸업.
1986년~2001년 LG전자 책임연구원. 1999년~현재 창원대 전자공학과 박사과정.
※관심분야: 시스템 제어 및 모터제어 응용.

박승엽(Seung-Yub Park)

1981년 고려대 전기공학과 졸업.
1984년 고려대 대학원 전기공학과 석사졸업.
1988년 고려대 대학원 전기공학과 박사 졸업.
1988년~현재 창원대학교 전자공학과 교수.
※관심분야: 플랜트 제어 및 고장진단, 모델링.