

論文99-36C-7-3

인터럽트 발생기를 사용한 접속 비트 전환식 양방향 접속장치의 설계

(Design of Bit Selectable and Bi-directional Interface Device using Interrupt Generator)

林 泰 榮 * , 李 天 熙 **

(Tae-Young Lim and Cheon-Hee Yi)

요 약

본 논문에서는 접속된 주변장치와 데이터를 비동기로 교환 할 수 있는 접속 비트 전환식 양방향 접속 장치 설계에 대하여 기술한다. 특히 인터럽트 발생기를 설계하기 위해서 제시된 펄스형 순차회로 파형을 합성 할 수 있는 “진리치 비교 알고리즘”을 제안하고 이를 인터럽트 레지스터 설계에 적용한 것에 대하여 상술하였다. 또한 개별 비트 출력 신호들의 클럭 스큐 현상을 없애기 위한 “최종 출력 핀 제어” 기법을 시스템에 적용하였으며 나머지 블록들을 설계한 기법에 대하여도 상술하였다. 이러한 기법을 사용함으로써 본 논문의 접속 포트들은 0.7ns 이내의 지연시간을 나타내었다.

Abstract

In this paper, Bit selectable and Bi-directional Interface Device is described, which can communicate data with the peripheral devices. Specially, an algorithm of “truth-table comparison” that synthesizes the pulse-type sequential circuit pulse has been proposed to design the Interrupt Generator, and implemented in designing the Interrupt Register. Also, a description of the asynchronous design method is given to remove the clock skew phenomenon, and the output asynchronous control method which finds the optimal clock and controls all the enable signal of the output pins at the same time is presented. Using this technique interface ports have delay time of less-than 0.7ns.

I. 서 론

본 논문은 TV 브라운관과 컴퓨터 모니터에 사용되는 마그네트(Magnet)에 일정한 자력을 갖도록 자화시키는 착자기를 제어하며, 검사공정을 자동화하는 착자 자동화 제어 시스템용의 집적회로 일부를 설계하여 개발한 것에 관한 것이다^[1].

착자 자동화 제어 시스템은 공장 자동화 제어 시스템의 일종으로써 CPU, RAM, ROM, PPI 및 키보드 입력부, LED 표시부, 자력 출력부, 자기 입력부 등의 주변장치로 구성 되어있다. 착자 자동화 제어 시스템의 주변장치들을 제어하기 위해서는 72비트의 양방향 접속 포트가 필요하고, 각종 주변장치들의 다양한 요구 사항을 수용 할 수 있는 접속 회로가 필요하다. 따라서 접속 포트의 비트 수를 가변 할 수 있는 구조의 설계와 주변 장치와 교신하는 핀들의 역할을 전환시킬 수 있는 구조로 설계되어야 한다. 또한 키보드를 눌렀을 때 그 정보가 즉각 시스템에 전달하도록 하려면, 주변장치의 입력 또는 상태 신호들으로써 시스템 CPU에 직접 인터럽트(Interrupt)를 요청할 수 있고, CPU의 인터럽트 응답을 받아들여서 측근의 주변장치를 직

* 正會員, 韓國電子通信研究院

(ETRI)

** 正會員, 淸州大學校 電子工學科

(Department of Electronic Engineering, Chongju University)

接受日字:1999年5月25日, 수정완료일:1999年6月28日

접 제어 할 수 있게 하는 타이밍 전략이 필요하며, 이는 주변장치와 교신하는 접속 포트 부분을 비동기 식으로 설계함으로써 해결의 한 방법이 될 수 있고, 동기 식으로 설계한 것에 비해서 고속동작을 추구 할 수 있다.

한편, CMOS SOG 회로의 전압과 기생 캐패시턴스 값은 베이스 어레이(Base array)와 디자인 룰(Design rule)에 따라서 특정한 값으로 고정된다. 따라서 베이스 어레이와 디자인 룰이 확정된 CMOS SOG 회로에서 전력소모를 줄이는 방법은 주어진 일에 필요한 게이트 출력의 충전과 방전의 회수를 줄이는 방법을 모색하는 것이 유일한 방법이다^[2]. 이는 주 클럭(Main clock)을 사용하지 않는 데이터-흐름 회로 구조를 고려 할 수 있다. 이 구조는 여러개의 상호치리 코어가 있는 데이터-흐름 아키텍처를 사용한다. 이 코어는 클럭을 사용하지 않고, 대신에 명령이나 데이터의 도착에 근거하여 명령을 실행한다. 코어에서 클럭을 없앴다는 것은 명령이나 데이터가 없을 때 IC가 전력을 거의 사용하지 않으므로 저전력 소자를 구현할 수 있게 된다^[3].

이러한 배경으로 CMOS SOG(Sea Of Gate)용 범용 접속기(Common Interface)로써 고속/저전력 동작에 적합한 프로그래머블 주변장치 접속기(PPI : Programmable Peripheral Interface)를 설계하였으며^[4,5], 본 논문에서는 주변장치와 접속하는 포트들의 비트 수를 가변 할 수 있으며, 접속 핀들의 역할을 전환시킬 수 있는 그림 1의 점선부분에 해당하는 “비트 전환식 양방향 접속장치”를 설계한 것에 대하여 기술하였다.

특히, 인터럽트 레지스터를 설계하기 위해서, 제시된 펄스형 순차회로 파형을 합성 할 수 있는 “진리치 비교 알고리즘”을 제안하고 이를 A1 인터럽트 레지스터 설계에 적용한 것에 대하여 상술하였다. 또한 개별 비트 출력 신호들의 클럭 스쿠(Clock skew) 현상을 없애기 위해서 이네이블(Enable) 신호 생성기의 출력을 래치(Latch) 시키는 “최종 출력 핀 제어” 기법에 대하여도 상술하였다.

본 논문에서는 비트 전환식 양방향 접속장치 1조에 국한하여 기술하였는데, II장에서는 비트 전환식 양방향 접속장치의 구성, III장에서는 C 포트 제어부의 설계, IV장에서는 진리치 비교 알고리즘을 적용한 인터럽트 레지스터의 설계, V장에서는 C 포트 접속 및 패

드부의 설계를 기술하고 VI장에서 결론을 기술하였다.

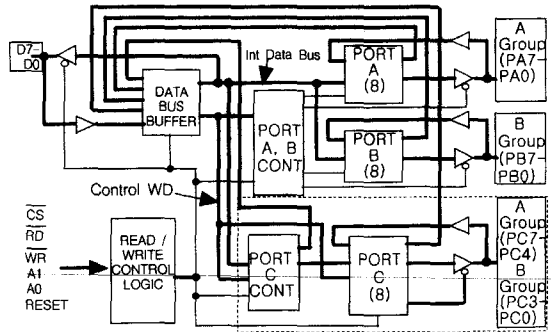


그림 1. PPI의 블록도
Fig. 1. Block diagram of the PPI.

II. 비트 전환식 양방향 접속장치의 구성

1. 비트 전환식 양방향 접속장치의 기능 사양

비트 전환식 양방향 접속장치는 사용자의 프로그래밍에 의해 입/출력 동작 모드를 설정 할 수 있도록 설계하여, 세가지의 중요한 동작 모드를 수행하도록 하였다. 0번 모드에서는 일반적인 접속기가 되도록 설계하고, 1번 모드에서는 입력과 출력 주변장치를 제어하기 위한 인터럽트 신호 입출력 용도가 되도록 설계하고, 2번 모드에서는 양방향 주변장치를 제어하기 위한 인터럽트 신호 입출력 용도가 되도록 설계하였다.

그림 2에 동작모드에 따른 각 핀들의 기능을 나타내었다. 그림에서처럼 PC0부터 PC7까지의 각 핀들은, 설정하는 동작 모드에 따라서 입력이나 출력 핀이 되기도 하고, 비트별로 인터럽트 관련 기능의 담당 역할 전환이 가능하므로 “비트 전환식 양방향 접속장치”로 표현하였다. 이 후 “양방향 접속장치”로 표기하기로 한다. 0번 모드의 작동은 매우 단순하므로 상세 설명은 생략하였다.

MODE 0			MODE 1			MODE 2				
	IN	OUT	IN	OUT	IN	OUT	GROUP A ONLY			
PC ₀	IN	OUT	PC ₀	INTR _B	INTR _B	PC ₀	I/O			
PC ₁	IN	OUT	PC ₁	IBF _B	OBF _B	PC ₁	I/O			
PC ₂	IN	OUT	PC ₂	STB _B	ACK _B	PC ₂	I/O			
PC ₃	IN	OUT	PC ₃	INTR _A	INTR _A	PC ₃	INTR _A			
PC ₄	IN	OUT	PC ₄	STB _A	I/O	PC ₄	STB _A			
PC ₅	IN	OUT	PC ₅	IBF _{BA}	I/O	PC ₅	IBF _{BA}			
PC ₆	IN	OUT	PC ₆	I/O	ACK _A	PC ₆	ACK _A			
PC ₇	IN	OUT	PC ₇	I/O	OBF _A	PC ₇	OBF _A			

그림 2. 동작 모드의 종합 정리 표
Fig. 2. Mode definition summary.

2. 양방향 접속장치의 구조 및 타이밍 사양 설정

양방향 접속장치는 1개의 C 포트 제어부, 1개의 C

포트 접속부 및 8핀의 양방향 입출력 패드부와 접속 핀부로 구성한다.

그림 3에 이 접속장치의 세부 블록도를 나타내었다. 그림의 좌측에 있는 신호들은 PPI 시스템 내부 블록과 연결되는 것이다. C 포트 제어부는 양방향 접속장치의 코아로써 비트 셀/리셀 및 데이터 저장기(Bit Set/Reset & Data REG) 1개와 인터럽트 생성 및 접속 전환기(인터럽트 발생기와 접속회로 전환기(MUX) 여러개 및 조합회로 #2)로 구성한다. 이 제어부는 C 포트의 상위 4비트와 하위 4비트의 출력 이네이블을 개별적으로 제어하면서, 데이터 버스 입력을 받아들여 비트 셀/리셀 및 데이터 저장 회로가 동작하도록 설계하며, 인터럽트에 관련된 신호들을 생성해서 입/출력 되도록 설계한다. C 포트 접속부는 버퍼 및 출력핀 제어 신호 생성용 조합회로 #1으로 구성한다. 패드부는 양방향 입출력 패드로 구성하고 우측에 주변장치와 접속하는 핀들을 나타내었다.

양방향 접속장치는 저전력 설계를 위하여 주 클럭(Main Clock)을 사용하지 않고, 대신에 명령이나 데이터의 도착에 근거하여 명령을 실행 할 수 있는 데이터-흐름 회로 구조를 갖도록 설계한다.

양방향 접속장치의 데이터-흐름은 다음과 같다. 그림 3의 좌측에 있는 비트 셀/리셀 겸 데이터 저장기와 접속회로 전환 제어용 조합회로 #2에는 연결단자를 통하여 8비트의 데이터와 8비트의 콘트롤 워드(명령어)가 도착하고 이들과 동반하는 쓰기(WR)나 읽기(RD) 신호가 도착한다. 이후 쓰기나 읽기 신호의 상승시점(라이징 엣지, Rising Edge)을 감지하여 이 때 저장기와 조합회로 #2가 작동, 출력한다.

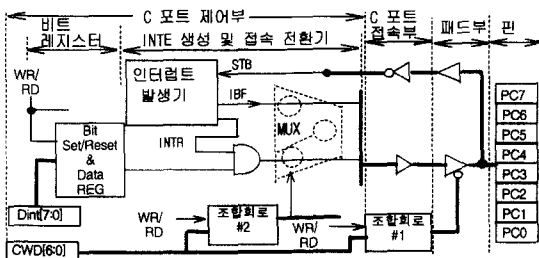


그림 3. 비트 전환식 양방향 접속 장치
Fig. 3. Bit selectable & Bi-directional Interface Device.

또한, 양방향 접속장치는 주변장치로부터 입력되는 신호에 대하여 실시간(Real Time)으로 동작 할 수 있도록 비동기 형으로 설계한다.

양방향 접속장치가 입력담당 주변장치의 작동 요청에 의거하여 수행하는 실시간 동작은 다음과 같다. 그림 3의 우측에 있는 외부 접속 핀으로 주변장치(예, 키보드)가 눌러졌음을 알리는 스트로브(Strobe) 신호가 인터럽트 발생기에 입력된다. 이 신호는 읽기나 쓰기 유무와 관계없이 즉각적으로 인터럽트 발생기를 구동 시켜서 입력 버퍼 찻음(IBF, Input Buffer Full) 신호를 외부 접속 핀으로 출력해서 주변장치에게 받아들일 준비가 됐음을 알림과 동시에 CPU에게 인터럽트 요청(인터럽트 리퀘스트, INTR) 신호를 출력하도록 한다. 이 후에 CPU로부터 오는 읽기 명령에 근거하여 인터럽트 금지(Interrupt Disable)를 시키고 입력 버퍼 찻음 신호를 종료시킨다.

이와같은 각종 신호들의 흐름과 작동방법은 1번 모드로 작동하는 것이며, 이 모드에서의 입력담당 주변장치의 스트로브 신호에 근거한 상세 파형 사양을 그림 4로 나타내었다. 그림에서 점선으로 표시한 부분이 양방향 접속장치의 출력으로써 이와 동일한 파형들이 출력되도록 설계한다.

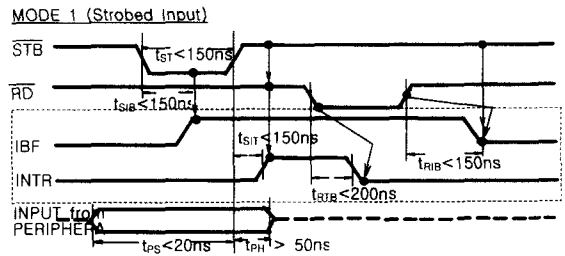


그림 4. 입력담당 주변장치의 상세 파형 사양
Fig. 4. Strobbed Input Wave form.

III. C 포트 제어부의 설계

1. 비트 셀/리셀 및 데이터 저장기의 기능

PPI의 콘트롤 워드를 분석하면, 비트 셀/리셀 형식은 그림 5와 같은 형식으로 되어있다^[4].

그림 5는 그림 6의 A0 입력이 "1"인 상태일 때, 다시 말해서 C 포트를 지정하거나 콘트롤 워드 임을 지정한 상태 일 때, 데이터 버스(또는 콘트롤 워드)의 D7이 "0"이면 이 버스 데이터를 "비트 셀" 모드의 콘트롤 워드로 동작시키겠다는 것을 의미하고, 이 때 비트 셀 모드의 콘트롤 워드는 하위 4비트(D3, D2, D1, D0)를 디코딩하여 PC0부터 PC7까지 8비트 중의 하나를 선택해서 "셀" 하고 D0로 이들을 "리셀" 시킨다.

디코딩하는 방법을 좀더 살펴보면, D3, D2, D1이 "101"이고 이 때 D0 이 "1"이면 PC5 비트를 쉘 하고, 이 때 D0 이 "0"이면 PC5 비트를 리셀 한다. 따라서 이 부분은 D3, D2, D1 세 비트를 3-8 디코더로 디코딩하고 D0로 이를 "이네이블" 시키거나 "디스에 이블" 시키는 회로가 필요하다.

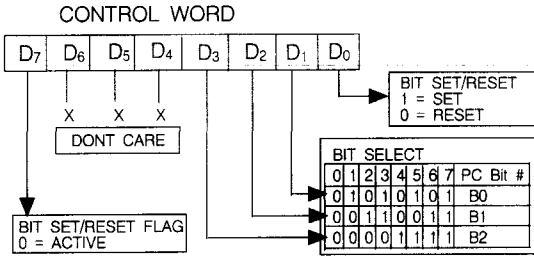


그림 5. 비트 쉘/리셀 형식
Fig. 5. Bit Set/Reset Format.

한편, A0 입력이 "0"인 상태 일 때는 이 버스 데이터를 내부 데이터로 간주하여 데이터 저장기를 통해 C 포트 접속부로 출력한다. 따라서 이 부분은 A0 신호로 제어하는 MUX를 구성, 비트나 데이터를 출력하고 종단에 플립플롭으로 저장할 수 있는 회로가 필요하다.

2. 비트 쉘/리셀 및 데이터 저장기의 세부 설계와 시뮬레이션

비트 쉘/리셀 및 데이터 저장기는 비트 쉘 디코더와 비트 리셀 디코더 각 1개, 8비트의 접속회로 전환기 (MUX) 1개, 8비트의 D F/F 1개 및 기타 논리소자들로 구성한다.

비트 쉘 디코더 회로는 8비트의 데이터 버스 신호 중에서 하위 세 비트(D3, D2, D1)가 입력되면, 항상 PC0부터 PC7까지의 접속 포트 중 하나를 디코딩하여 쉘 되도록 설계한다. 비트 리셀 디코더 회로는 데이터 버스 신호중에서 하위 네 비트(D3, D2, D1, D0)가 입력되면, 항상 PC0부터 PC7까지의 접속 포트 중 하나를 디코딩하여 리셀 되도록 설계한다. 8비트의 접속 회로 전환기(MUX)는 어드레스 A0가 "0" 일 때는 데이터 버스 신호를 선택하고, 어드레스 A0가 "1" 일 때는 디코더 된 결과인 C 포트의 1 비트를 쉘 한 값을 선택하도록 설계한다. 8비트의 D F/F은 접속회로 전환기의 값을 받아들여 저장, 출력하되 "레지스터 리셀(P1)" 신호로 클리어 되도록 설계한다. 이 부분은 LODECAP^[6]의 회로도 입력기(Schematic Capture)

로 설계하였고, 8비트의 D F/F을 포함하여 완성된 블록의 구성도는 그림 6에 나타내었다.

그림에서 D0가 "1" 일 때는 비트 쉘 디코더의 값이 출력되며, D0가 "0" 일 때는 비트 리셀 디코더의 값이 MUX를 거쳐서 인버팅 된 상태로 NOR 게이트에 동시에 인가되므로 비트 쉘/리셀 레지스터는 C 포트를 리셀 시키는 "0"의 값들을 출력한다. 그림 7은 이의 시뮬레이션 결과이며, CD(7:0)의 하위 4비트에 의해 C 포트의 각 비트를 쉘 시키거나 리셀(00) 시킨을 확인 할 수 있다.

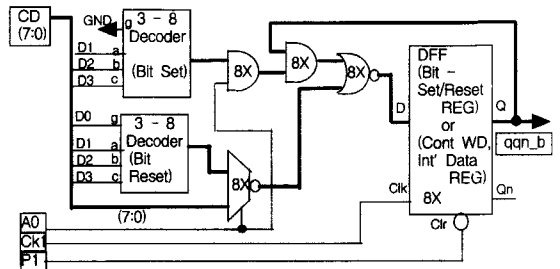


그림 6. 비트 쉘/리셀 레지스터 블록도
Fig. 6. Bit Set/Reset Register Block diagram.

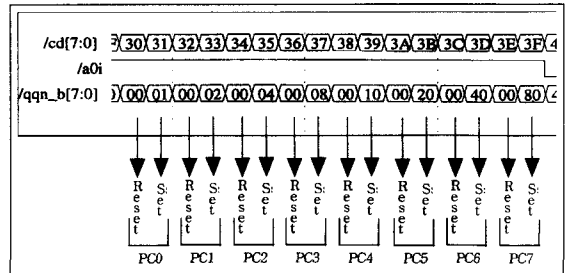


그림 7. 비트 쉘/리셀 레지스터의 Sim 파형
Fig. 7. Sim. Result of Bit Set/Reset Register.

3. 인터럽트 생성 및 접속 전환기의 개별 비트 제어를 위한 전환 구조

인터럽트 생성 및 접속 전환기는 인터럽트 발생기, 접속회로 전환기(2:1 MUX)와 조합회로 #2로 구성하고, 콘트롤 워드로 설정되는 0번 모드에서 8 비트의 내부 데이터를 출력하고, 1번과 2번 모드에서 일부 비트들은 데이터를, 나머지 비트들은 인터럽트 관련신호를 출력 할 수 있도록 설계한다. 따라서, 인터럽트 생성 및 접속 전환기의 일부 출력들은 그림 3에서 점선으로 표시한 그림 9와 같은 접속회로 전환(MUX) 구조를 갖는다.

그림에서 인터럽트 생성 및 접속 전환기의 출력은 C

포트 접속부의 입력이 되는 것으로써, 전단에 있는 2:1 MUX로 0번 모드와 1, 2번 모드 일 때를 제어하여 그림 좌측의 비트 셀/리셀 및 데이터 저장기의 출력이나 인터럽트 발생기의 출력신호중 한 가지를 선택하여 출력하며, 이 제어는 콘트롤 워드의 조합으로 수행된다. 이 때, 1번 모드에서는 A와 B 인터럽트 레지스터가 존재하며, 2번 모드에서는 1번과 2번 인터럽트 레지스터가 존재한다. 1번과 2번 모드에서 동작하는 인터럽트 레지스터들과 조합회로들을 통칭하여 인터럽트 발생기라 한다. 인터럽트 신호를 담당하는 비트들은 이와 같은 전환 구조를 갖도록 설계한다.

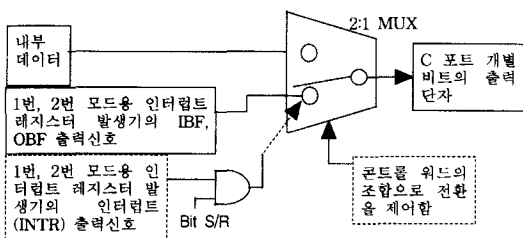


그림 8. 접속 전환기의 전환 구조
Fig. 8. Conversion Structure.

4. 인터럽트 생성 및 접속 전환기의 세부 설계

양방향 접속장치의 최종 입/출력단자는 8 비트의 주변장치 접속용 포트이며, 1번 동작 모드에서 입력용 주변장치의 입력을 받아들이기 위하여 PC6과 PC7, 두 비트를 양방향 데이터 비트로 할당하고, PC4와 PC2, 두 비트로는 스토브 신호를 입력한다. 또한 출력용 주변장치에 출력 하기 위하여 PC4와 PC5, 두 비트를 양방향 데이터 비트로 할당하고, PC4와 PC2, 두 비트로는 액nowled지(ACK) 신호를 입력한다. 또한 2번 동작 모드에서는 양방향 주변장치의 입력을 받아들이기 위하여 PC2와 PC1 및 PC0 세 비트를 양방향 데이터 비트로 할당하고, PC4 비트로는 스토브(STB) 신호를 PC6 비트로는 액nowled지(ACK) 신호를 입력한다. 결국에 1번과 2번 동작 모드에서 공통으로 할당하게 되는 입력용 비트는 PC2, PC4, PC6 이상 세 비트들이며, PC0, PC1, PC3, PC5, PC7 이상 다섯 비트들은 1번 또는 2번 동작 모드에서 출력 형태의 신호들로 할당된다. 이 출력 형태의 신호들은 인터럽트 생성 및 접속 전환기의 출력이다. 따라서, 접속 회로 전환기(2:1 MUX)는 PC0, PC1, PC3, PC5, PC7 이상 다섯 비트에 대하여 데이터 또는 인터럽트 관련 신호들을 선택, 전환하도록 설계해야 한다. 이 때

어떤 신호를 선택하는가 하는 것은 콘트롤 워드로 수행되고 위 다섯 비트들의 인터럽트 관련 신호들은 A, B, 1, 2 인터럽트 발생기의 출력 신호들이다.

이상을 정리하면 인터럽트 생성 및 접속 전환기는 1번과 2번 동작모드용 A, B, 1, 2 번 인터럽트 발생기의 출력들 또는 내부 데이터를 C 포트 접속부로 출력한다. 1번 모드용 A 인터럽트 발생기의 값은 PC7, PC5, PC3 비트로 출력되며, 이의 전환 제어용 진리치는 표 1 처럼 정리 할 수 있다.

표 1. A 인터럽트 전환 제어용 진리치표
Table 1. Truth table of the A INTE.

C 포트 전환 제어용 콘트롤 워드			주변장치의 종류		C 포트 전환용 MUX 제어 출력			비고
Y6	Y5	Y4	출력용 PC[6] (WR)	입력용 PC[4] (RD)	PC[7] 전환	PC[5] 전환	PC[3] 전환	
0	0	0	1	1	1	1	1	모드 0
0	0	1	1	1	1	1	1	모드 0
0	1	0	0	1	0	1	0	모드1, PA출력
0	1	1	1	0	1	0	0	모드1, PA입력
1	X	X	0	0	0	0	0	모드 2

표 에서 주변장치의 종류에 입력용이라고 표기한 것은 키보드 종류의 주변장치가 숫자 데이터를 넣겠다는 것을 미리 알리는 STBA(Strobe Input A INTE) 신호이며, 콘트롤워드의 Y6, Y5, Y4 비트가 "0 1 1" 또는 "1 X X" 일 때 "Low" 에 해당하는 난의 출력 즉 신호중 "0"을 나타낸 비트(PC5 와 PC3 또는 PC7 PC5 PC3)가 인터럽트 관련 신호들을 선택, 출력하는 것이다.

이 진리치표를 구현하는 조합회로 #2 부분을 LODECAP^[6]의 진리치 편집기(Truth table editor)로 설계하였다. 접속 포트 PC7, PC5, PC3 비트에 대한 A 인터럽트 전환체계를 나타내면 그림 9와 같다. 그림에서 스위치로 표시한 것은 2:1 MUX를 의미하며 표 1에 나타난 C 포트 전환 제어용 콘트롤 워드에 의해 전환이 제어된다.

또한, 0번 모드에서는 전환 위치가 전부 위쪽에 접속되고 내부 데이터를 선택 출력한다. 1번 모드의 경우에는 전환 위치가 전부 아래쪽으로 접속되고, 그림 4의 입력담당 주변장치의 상세 파형과 같이, 입력용 주변장치에서 PC4로 입력되는 STBA(주변장치가 데이터를 넣겠다는 것을 미리 알려주는 신호)가 인터럽트 레지스터(A1)를 동작시켜서 IBF를 PC5로 출력하고 읽기 신호로 IBF 신호를 리셀 시킨다. 또한

PC3을 통하여 인터럽트 요청(INTR: Interrupt Request) 신호를 외부로 출력한다. 이 때 인터럽트 요청 신호는 비트 셀/리셀 및 데이터 저장기의 출력에 의해 셀/리셀 된다. PC0와 PC1의 출력들도 이와 동일한 방법으로 설계한다.

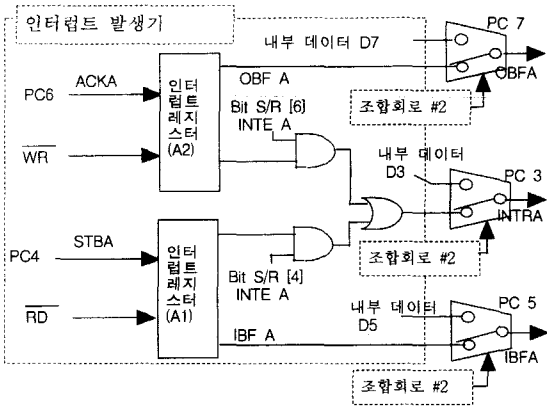


그림 9. A 인터럽트용 전환 체계도
Fig. 9. A INTE Conversion Structure.

IV. 진리치 비교 알고리즘을 적용한 인터럽트 레지스터의 설계

1. A1 인터럽트 레지스터의 필요 파형

인터럽트 레지스터는 인터럽트 발생기의 구성 요소이다. 이 부분은 그림 4와 같은 파형을 생성하도록 설계 해야하며, 2절에서 밝힌 바와 같이 주 클럭(main clock)이 없는 비동기 형으로 설계한다.

인터럽트 레지스터는 그림 9에 나타낸 것을 포함하여 A1, A2, B1, B2, 1, 2 이상 6개가 있다. 우선 입력담당 장치용 A1 인터럽트 레지스터를 살펴보면, A1 인터럽트 레지스터는 그림 4의 점선 부분과 같이 주변 장치로부터 스트로브 입력을 받아서 IBF를 출력하고, 읽기(RD) 신호의 상승 엣지에서 IBF를 종료 해야한다. 그림 4에서 STB 신호는 레벨 입력(Level Input) 이고, RD는 엣지입력(Edge Input)이다. 따라서 이 부분은 레벨 신호와 펄스 신호들이 혼합된 순차회로로 설계해야 한다. 또한 그림에서 INTR 출력도 STB와 RD 입력에 의해 출력한다.

비동기 펄스형 순차회로 설계 알고리즘 중에 마스터-슬레이브 플립플롭 배열을 사용하는 기법이 있다^[7]. 이 방법은 사용할 수 있는 플립플롭의 종류가 이미 확정된 0.8um SOG에 적용하기가 용이하지 않다. 따라

서, 본 논문에서는 “진리치 비교 알고리즘”이라고 명명한 “제시된 펄스형 순차회로 파형을 합성 할 수 있는 알고리즘”을 제안하고 이를 A1 인터럽트 레지스터 설계에 적용한다.

2. 진리치 비교 알고리즘의 제안 및 A1 인터럽트 레지스터 설계 적용

진리치 비교 알고리즘은 제시된 파형을 구현할 수 있는 펄스형 순차회로 합성 알고리즘으로써, “레벨형 입력과 펄스형 입력이 혼합된 비동기 회로에서 사용하려는 플립플롭을 설정한 후 파형 동작 순서에 따라서 신호와 매칭 시킬 단자를 가정하여 진리치를 대입한 후, 가정한 단자의 논리 방정식을 구현, 회로를 설계” 할 수 있는 알고리즘으로써, 단계별 구현 방법과 IBF 신호용 인터럽트 레지스터의 설계 예는 다음과 같다.

[단계 1] 구현 대상 파형의 입력신호를 레벨 신호와 펄스 신호로 구분하고 1개의 출력 신호를 지정한다.

A1 인터럽트 레지스터는 스트로브 입력이 인가된 후 종료 이전에 IBF가 천이 되므로 스트로브를 레벨 신호로, 읽기 신호의 라이징 엣지에서 IBF를 천이 시켜야 하므로 읽기 신호를 펄스 신호로 구분하고, 출력 신호로 IBF를 지정한다.

[단계 2] 적용할 플립플롭을 설정하고 주어진 입력과 동일한 조건의 단자를 선정하며, “이전상태 또는 언-노운(Unknown) 상태”를 피할 수 있는 초기화 신호를 제시된 파형의 앞쪽에 부가한다.

A1 인터럽트 레지스터는 0.8u SOG로 구현하므로 이 디자인 룰에서 가장 보편적으로 사용하는 “dfbfnb(D Flip Flop with Set and Clear)”를 적용할 플립플롭으로 설정하고 읽기(RD)를 펄스 신호로 CP 단자에, 스트로브를 레벨 신호로 CDN 이나 SDN 에, 무관(Don't care) 설정이 많은 D 는 Vcc나 Vss 에 접속시킬 것으로 예정하며, 초기화 신호를 결정한다. 이 때는 이전(q, qn) 또는 언-노운(Unknown) 상태를 제외하고, 라이징 행을 제외하며, 출력이 “0”으로 안정된 상태를 선정한다. 따라서 그림 10의 (b)에서 1행과 7행이 남게 되는데, 마지막으로 입력신호의 성격을 고려하여 스트로브 신호가 “1”상태가 될 수 있는 1행을 초기화로 선정하여 CP를 “0”으로 설정한다. 본 예제에서는 그림 11 처럼 “1”과 “0”을 초기화 신호로 덧붙인다.

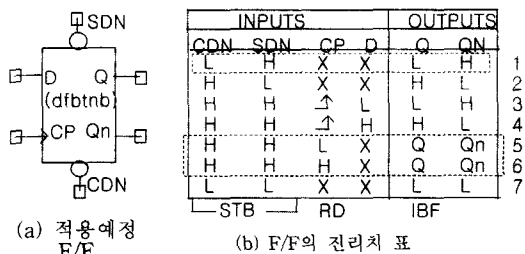


그림 10. F/F와 진리치 표
Fig. 10. F/F & Truth table.

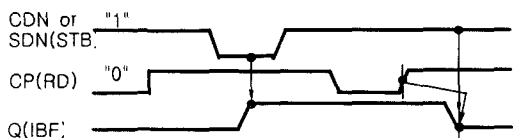


그림 11. 초기화 신호 부가 파형
Fig. 11. Initialized Waveform.

[단계 3] 제시된 파형의 동작 순서를 정하고 진리치표를 만든다. 이 때 천이가 발생하는 기점을 한 구간으로 설정한다.

A1 인터럽트 레지스터는 초기화를 포함하여 6개로 동작을 구분 할 수 있으며, 이 때 천이가 발생하는 구간을 기준으로 구분한다. 실제 설계 시에도 이처럼 지연 시간을 고려하지 않고 있다. 구분한 결과는 그림 12(a)와 같고 a, b, c, d... 순으로 동작하는 것으로 재설정할 수 있으며, 동작 순서의 진리치 표는 그림 12(b)로 나타내었다.

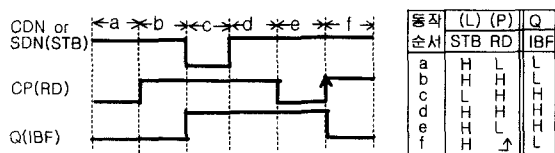


그림 12. (a)파형 수정, (b)동작순서의 진리치 표
Fig. 12. (a)Waveform modify, (b)Opt sequence.

[단계 4] 진리치 비교표를 만들고 파형 동작순서 진리치 행과 동일한 적용 예정 플립플롭의 진리치 행을 비교하여 대입하며, 유사한 행들은 레벨 신호와만 비교하여 할당한다.

이 때 한 동작 순서에서 발생 가능한 진리치들은 전부 대입하며, 대입 순서는 펄스신호 입력 난부터 다음 차례로 대입한다. 펄스 신호 입력 난을 우선으로 하는 이유는 이 신호가 가장 안정적이며, 확실한 진리치들을 나타내기 때문이다. 또한 동일한 난에서 대입하는

순서는 동작순서의 진리치 표로 확정된 값 중에서 출력 값부터 좌측방향으로 대입한다. 대입한 결과는 그림 13과 같다. 그림에서 어떤 단자가 한 칸에서 2개의 진리치가 있는 경우에는 여하한 값을 지정하든지 무관하며, 이 남은 점선 타원형으로 표시하였다.

대입 순서	STB		RD	D	Q	dfbtb 번호
	CDN	SDN	CP		IBF	
1(f)	H	H	↑	L	L	3
2(a)	L	H	X	X	L	1
3(b)	L	H	X	X	L	5
4(c)	L	H	X	X	L	1
5(d)	H	L	X	X	H	6
6(e)	H	L	X	X	H	2
	H	H	X	X	H	6
	H	L	X	X	H	2
	H	H	L	X	Q	5

그림 13. 진리치 비교표
Fig. 13. Truth Value Comparison Table.

[단계 5] 진리치 비교표의 CDN과 SDN 단자에서 2개의 논리 값이 존재하는 동작순서를 제외한 나머지 동작 순서의 논리 값에 "1" 또는 "0"을 마스크 하였을 때 모든 경우의 합산 값이 "0"인 경우의 단자와 마스크 값을 구한다.

- (1) CDN 단자에 전부 "1"을 가 하였을 경우
 $CDN(f) XOR H + CDN(c) XOR H + CDN(d) XOR H + CDN(e) XOR H = "0"$ (성립)
- (2) SDN 단자에 전부 "1"을 가 하였을 경우
 $SDN(f) XOR H + SDN(a) XOR H + SDN(b) XOR H + SDN(c) XOR H = not "0"$ (성립 안 됨)
- (3) CDN 단자에 전부 "0"을 가 하였을 경우
 $CDN(f) XOR L + CDN(c) XOR L + CDN(d) XOR L + CDN(e) XOR L = not "0"$ (성립 안됨)
- (4) SDN 단자에 전부 "0"을 가 하였을 경우
 $SDN(f) XOR L + SDN(a) XOR L + SDN(b) XOR L + SDN(c) XOR L = not "0"$ (성립 안됨)

결론 : CDN 단자에 전부 "1"을 가 하였을 경우 만 성립되므로 $CDN = 1$ 이다.

[단계 6] 나머지 단자들의 식을 구하고 회로도로 구현한다.

D 단자 = "0", Cp 단자 = RD, SDN 단자 = STB, 단계 5에서 구한 값 $CDN = 1$

3. INTR 신호용 회로 합성 알고리즘
INTR 신호 생성을 위한 진리치 비교 알고리즘은

단계 3의 경우에 그림 14와 같고 단계 6은 다음과 같다.

[단계 6] INTR 신호 생성을 위한 식을 구하고 회로도도로 구현한다.

D 단자 = "1", Cp 단자 = STB, CDN = RD, 단계 5에서 구한 값 SDN 단자 = 1

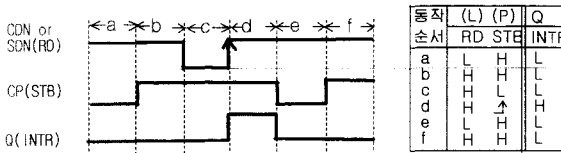


그림 14. (a)파형 수정, (b)동작순서의 진리치 표
Fig. 14. (a)Waveform modify, (b)Opt sequence.

4. A1 인터럽트 레지스터의 구현 회로

최종적으로 진리치 비교 알고리즘을 각각 적용한 IBF와 INTR 신호용 A1 인터럽트 레지스터 회로는 그림 15와 같다. 그림에서 좌측 플립플롭 부분이 IBF용이고 우측 부분이 INTR용이다.

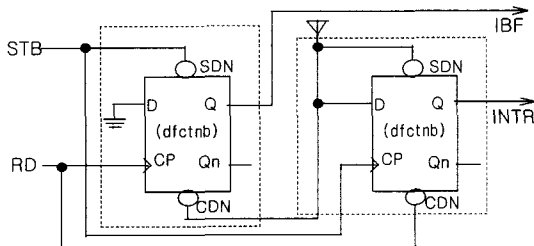


그림 15. A1 인터럽트 레지스터
Fig. 15. A1 Interrupt Register.

5. 진리치 비교 알고리즘의 분석

진리치 비교 알고리즘은 디자인 룰이 확정된 D-F/F을 적용하여 펄스신호와 레벨신호가 혼합된 비동기 순차회로를 설계하는데 적합하며, 신호에 대한 정의가 뒤바뀔 경우에도 그에 알맞은 결과를 나타낸다. 이는 다양한 파형에도 적용이 가능함을 의미하며, IBF와 INTR의 적용예에서 증명하였다.

또한, 진리치 비교 알고리즘은 출력신호의 파형을 정확하게 예측할 수 있다. 이는 진리치 비교표에서 알 수 있으며, 그림 16의 IBF용 알고리즘 분석으로써 이를 증명한다. 그림에서 대입순서 2, 3, 5, 6은 각각 2종의 진리치가 있다. 이 진리치는 단계 5의 결과에 따라 표 우측에 나타난 동작순으로 파형이 진행된다. 따라서 출력은 이 동작순에 따라 나타나며, 제 5열의 동작순 1을 보면 IBF의 값이 Q이다. 이는 최초에 시

작되는 파형이 언-노운(Unknown) 상태임을 의미하며, 이것으로써 IBF의 처음 값은 시뮬레이션 시에 논리적으로 중간 값을 표시할 것이라는 것을 예측할 수 있다.

또한, 진리치 비교 알고리즘은 JK 플립플롭에 대해서도 동일한 결과를 나타낸다.

대입 순서	STB		RD	D	Q IBF	dfctb 번호	동작순
	CDN	SDN	CP				
1(f)	H	H	X	L	L	3	← 6
2(a)	L	H	X	X	L	1	← 1
3(b)	L	H	X	X	L	1	← 2
4(c)	H	L	X	X	H	2	← 3
5(d)	H	L	X	X	H	2	← 4
6(e)	H	L	X	X	H	6	← 4
	H	H	X	X	Q	5	← 5

그림 16. 알고리즘 분석
Fig. 16. Algorithm Analysis.

6. 진리치 비교 알고리즘의 시뮬레이션 결과

진리치 비교 알고리즘으로 설계된 A 인터럽트 레지스터는 LODECAP(Logic DEsign CAPture) 툴의 회로편집기에서 회로도를 입력하고 파형편집기로 시뮬레이션을 수행하였으며 그림 17과 같은 결과를 얻었다. 그림에서 출력 IBF가 예측하였던 바와 같이 언-노운(Unknown) 상태임을 확인 할 수 있었다. 이 언-노운(Unknown) 상태는 최종회로에서 파워는 리셀 신호에 의한 초기화로 없앨 수 있었다.

한편, 인터럽트 생성 및 접속 전환기에 필요한 모든 인터럽트 발생기들은 전부 이 알고리즘으로 구현하였으며, 구현된 칩에서 50MHz의 동작속도로 완벽하게 동작함을 확인하였다.

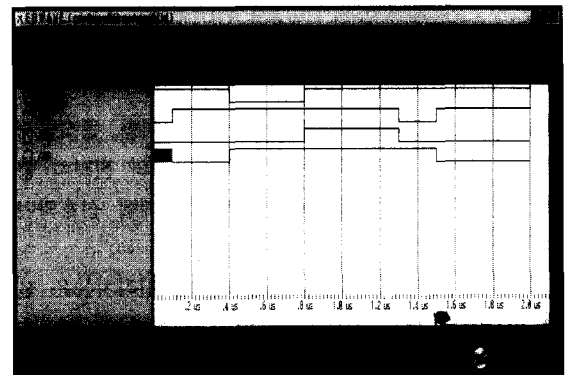


그림 17. 인터럽트 레지스터의 시뮬레이션
Fig. 17. Simulation result of the INTE.

V. C 포트 접속 및 패드부의 설계

1. C 포트 접속 및 패드부의 구성 및 설계

이 블록은 주변장치에서 인가하는 8-비트의 입력을 버퍼링하여 양방향 접속장치의 외부로 전송하고, C 포트 제어부에서 인가하는 신호를 받아들여 버퍼링 한 후 양방향 C 포트 패드의 출력측에 내 보내며, 콘트롤 워드 레지스터의 내용을 읽어들이어서 C 포트 가 출력 모드로 될 때를 선별, 이네이블 신호를 생성 시켜서 양방향 C 포트 패드의 출력 이네이블 단자를 제어하는 기능을 갖도록 구성한다.

2. 이네이블 신호 생성기의 세부 설계

PPI의 콘트롤 워드는 30개가 있으며, 콘트롤 워드에 의하여 모드 설정이 되고, 포트들의 입출력을 정의한다. 이네이블 신호 생성기(조합회로 #1)는 각 콘트롤 워드에 따른 C 포트의 출력 이네이블 논리와 모드 설정을 진리치표로 만들어서 “진리치표 합성기”로 설계하였다.

3. C 포트 접속 및 패드부의 세부 설계

이 블록의 설계에서는 송신 레지스터를 구동시키는 “읽기” 나 “쓰기” 신호에 맞추어 입출력 핀을 통해 출력되는 신호들의 타이밍을 최종 단에서 제어하는 “출력 신호 동시 제어 방식”을 제안한다.

기존의 PPI의 내부에 있는 양방향 접속 포트 중에서 출력용 주변장치에 대한 제어부분은 그림 18에서 D-Latch 부분이 없는 상태와 같은 구조로 되어 있다^[4]. 그림에서 각 출력 비트마다 전환회로와 조합회로 #1을 거치는 제어소자의 경로가 다르므로 OBF 출력 핀에서 약 10ns 정도의 지연 시간차가 발생하게 된다. 이는 비트별로 프로파게이션 지연이 다르게 나타나는 현상으로써 Top 회로를 구성 한 후 시뮬레이션 과정에서 슈퍼싱크 체크 에러가 발생하게 되는 원인이 된다.

4. 출력신호 동시 제어 방식에 의한 접속 포트의 설계

내부 데이터와 인터럽트 레지스터의 출력 신호는 각 비트별로 서로 지연 시간이 다르게 전환회로와 출력 핀에 입력될 수 있다. 이와 같이 여러 곳에서 지연을 일으킬 수 있는 경우에는 내부 블록 설계 시에 해결하는 것보다는 톱(Top) 회로에서 다루어져야한다. 따라

서 두 신호를 제어하는 핵심 신호를 찾아야 하는데 이 경우에는 읽기나 쓰기 신호이며, 이 신호가 조합회로 #1, #2의 최종 출력을 이네이블 시키는 역할을 담당하고 있다. 본 장치는 결국 읽기나 쓰기 신호의 라이징 엣지에서 전환회로의 제어신호를 구동하도록 함으로써 데이터와 인터럽트 레지스터의 출력 신호를 출력 패드의 제어단자 측까지 도달하게 하였다. 또한, 라이징 엣지를 검출하여 이 때부터 10ns 이후에 읽기나 쓰기 신호의 폭과 동일한 제 2의 읽기나 쓰기 신호를 만들어서 그림 18과 같이 조합회로 #1과 출력 제어 핀 사이에 삽입한 D-Latch를 구동 시켰다.

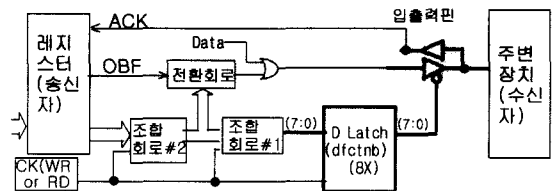


그림 18. 최종 출력 핀 제어 기법의 불력도
Fig. 18. Final Output Pin Enable Technic.

이를 “최종 출력 핀 제어(Final Output Pin Enable : FOPE)” 기법이라 하였고 이로 인하여 모든 비트들은 그림 19 처럼 동시에 0.7ns 이내의 지연 시간을 포함하며 출력 할 수 있었고, 클럭 스퀴(Clock skew) 현상을 제거할 수 있었으며, 슈퍼싱크 체크를 통과 할 수 있었다.

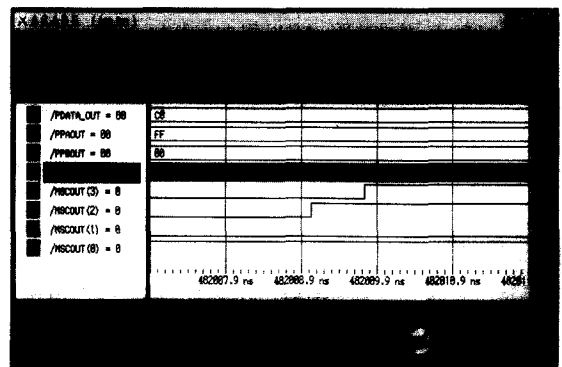


그림 19. 최종출력핀 제어기법의 시뮬레이션
Fig. 19. Simulation result of the FOPE.

5. 시뮬레이션 및 칩의 측정 결과

본 논문에서 제안한 비트 전환식 양방향 접속 포트를 포함하여 구현한 PPI의 최종 시뮬레이션 결과를 그림 20에 나타내었다. 그림의 PDATA_OUT은 이 장치에 인가되는 입력으로써, PAI가 “2” 일 때 인가

되는 "56"이, MSCOUT에는 "5"가 출력되고 LSCOUT으로 "6"이 출력되면서 4비트씩 나뉘어 출력되는 것을 확인하였고 완성된 칩에서도 동일한 출력을 얻을 수 있었다.

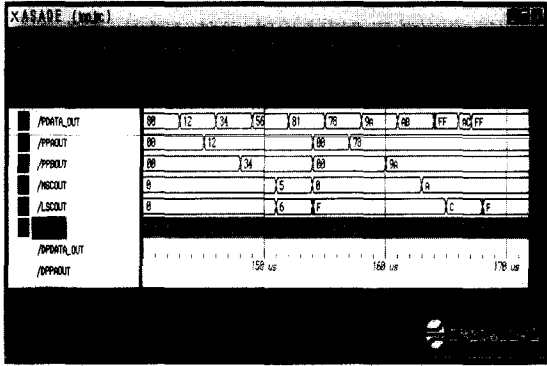


그림 20. PPI의 최종 시뮬레이션 결과 파형
Fig. 20. Final simulation result of the PPI.

VI. 결 론

본 논문에서 제안한 양방향 접속 포트를 LODE-CAP^[6]의 도면 편집기(Schematic Editor)와 진리치 합성기를 이용하여 회로도 입력과 블럭별 프리 시뮬레이션을 확인, 설계를 완성하였다. 또한 양방향 접속 포트가 포함된 PPI의 설계 및 시뮬레이션을 완료하고 ETRI의 반도체 공정라인을 이용하여 웨이퍼 가공 공정을 완료하였다. 개발된 ASIC은 50MHz에서 작동하고, 총 6,000 게이트로 ASIC화 되었다. 진리치 비교 알고리즘으로 구현한 인터럽트 레지스터를 포함하여

최종 출력 핀 제어 기법으로 설계한 출력 비트들 사이의 지연 시간차는 시뮬레이션에서 0.7ns 이내를 나타내었고 구현된 칩의 측정 결과와도 일치하였다. 본 연구 결과로 구현한 ASIC은 범용 주변장치 접속기의 용도로 폭넓게 적용할 수 있다.

참 고 문 헌

- [1] 이천희 외 1명, "차자자동화 시스템 및 통계분석틀 개발." 한국정보처리학회 제3권 4호, P.1014-1025, 1996년 7월
- [2] A. Bellaouar and M.I. Elmasry, "Low-power Digital VLSI Design." Kluwer Academic Publisher, 1995.
- [3] C.L. Seitz, "System timing." in Introduction to VLSI Systems, Addison-Wesley, 1980.
- [4] Pic 16/17 Micro controller data book, Microchip Technology Inc, 1995.
- [5] 임태영 외 3명, "자동차 및 검사자동화 시스템을 위한 집적회로 설계" 제6회 한국반도체 학술대회 논문집, pp.541-544, 1999년 2월
- [6] Y. H. Bae, etc. "VHDL Based ASIC Design CAD System Supporting Manual & Automatic Design", World Korean Scientist Workshop, July 1996.
- [7] Wakerly, J. F., "Digital Design Principles and Practices, Prentice-Hall, Englewood Cliffs, N.J., 1990.

저 자 소 개



李天熙(正會員)
1968년 한양대학교 전자공학과 졸업, 동대학원 졸업. 1975년 성균관대학교 대학원 전자자료처리학과 졸업. 1986년 성균관대학교 전자공학과 공학박사학위 취득. 1971년 ~ 1976년 한국마텔(주) 근무. 1977년 ~ 1978년 동양공업전문대학 전자공학과 근무. 1979년 ~ 현재 청주대학교 전자공학과 교수. 1983년 ~ 1985년 미국 캘리포니아 산호세 주립대학 객원교수. 주관심분야는 VLSI Layout, ASIC, DRAM, CAD Tool 개발 등



林泰榮(正會員)
1971년 한양대학교 공업경영학과 (학사). 1989년 한양대학교 산업대학원(전자공학 석사). 1994년 9월 ~ 현재 청주대학교 전자공학과 박사과정. 1971년 ~ 1974년 대한마이크로전자(주) 근무. 1974년 ~ 1978년 반도산업(주) 근무. 1978년 ~ 1981년 태평전자 근무. 1981년 ~ 현재 한국전자통신연구원 회로소자연구소 재직중. 주관심분야는 VLSI 설계, ASIC, Smart Power IC, IP 개발 등