

Slide 0

## 4장. 산업용 시스템버스 (VME 버스)

## VMEbus에 대하여

Slide 1

- Bus의 계층구조
  - 보드 레벨
    - \* 각 장치가 하나나 그 이상의 PCB로 구성된다
    - \* local bus 라고도 함
  - Backplane 레벨
    - \* 시스템을 구성하는 버스
    - \* IEEE Standard 1014 : VMEbus (1987)
    - \* IEEE Standard 1296 : Multibus II (1987년 인텔)
    - \* IEEE Standard 1196 : NuBus (1987년 MIT)
    - \* IEEE Standard 896.1 : Futurebus+ (1991)
  - Interface 레벨
    - \* I/O 장치나 I/O 채널을 위한 버스
    - \* SCSI, PCI, S 버스등

## VMEbus에 대하여 (Cont'd)

---

- Bus 의 선택사항

Slide 2

Options	High performance	Low cost
Bus width	Separate address and data lines	Multiplex address and data lines
Data width	Wider is faster (e.g., 64 bits)	Narrower is cheaper (e.g., 8 bits)
Transfer size	Multiple words have less bus overhead	Single-word transfer is simpler
Bus masters	Multiple (requires arbitration)	Single master (no arbitration)
Clocking	Synchronous	Asynchronous

## VMEbus에 대하여 (Cont'd)

---

Slide 3

- VMEbus (Versa Module Europe) 의 역사
  - 1979년 68K 프로세서를 위한 모토롤라의 Versabus 개발
  - 1981년 모토롤라, 모스텍 그리고 시스테틱스가 독일에서 Revision A 를 발표 (최초의 VMEbus)
  - 1982년 VMEbus 제조업자 그룹 (VITA: VMEbus Int. Trade Association) Revision B 발표
  - 1987년 IEEE1014 로 개선
  - 1989년 64Bit 의 Address와 Data를 지원하는 VME64 개발 (96핀, 80Mb/sec)
  - VME64는 ANSI/VITA1-1994 로 ANSI에 의하여 승인됨
  - 1996년 VME64x 표준안 발표, 표준안 진행중 (160 핀, 320Mb/sec)

## VMEbus에 대하여 (Cont'd)

---

- Slide 4**
- VMEbus의 특징
    - 마스터-슬레이브 구조
      - \* 다수개의 마스터-슬레이브가 존재가능
      - \* 다중프로세서와 다중프로세싱이 가능
      - \* 비동기식 데이터 전송사용
        - 다양한 전송속도를 가지는 보드구성 가능
        - 프로세서의 독립적인 구성가능
    - 8비트, 32비트, 64비트의 Address와 Data 전송가능

## VMEbus에 대하여 (Cont'd)

---

- Slide 5**
- VMEbus의 응용현황
    - 군사용, 통신용, 산업용에 다양한 분야에 응용됨
    - 초기 군사용, 현재 산업용 통신용으로 많이 사용됨
    - 군사용 응용: M1A1 탱크의 사격통제 장치
    - 산업용 응용: 아리안 로켓의 실시간 모니터링과 발사장치, 산업용 로봇 및 공작기계 제어
    - VMEbus의 응용이 늘고있는 이유
      - \* 각종 제어장치 및 프로세서의 제조업자가 달라도 쉽게 장착됨
      - \* 병렬제어, 분산제어가 가능
      - \* 실시간 OS를 사용하여 실시간 제어가 가능
      - \* VMEbus의 특허권이 없어서 로열티나 라이선스가 필요없음

## VMEbus의 구조 및 기능

- 시스템 버스의 일반적인 접속도

Slide 6

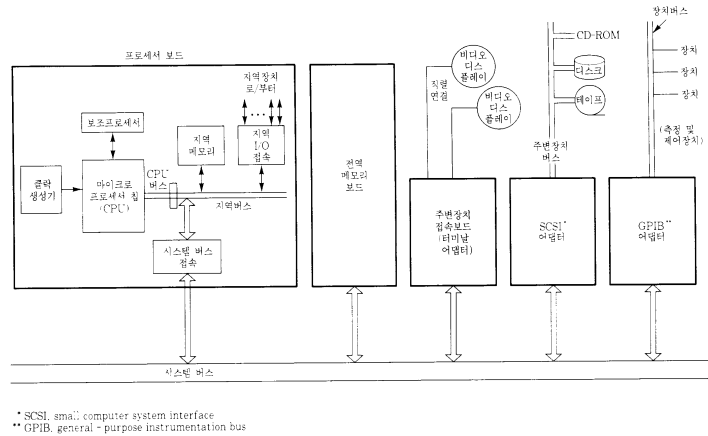


그림 4.1 시스템 버스를 통하여 연결된 프로세서, 메모리, I/O 및 주변 버스 접속 보드.

## VMEbus의 구조 및 기능 (Cont'd)

Slide 7

- VMEbus의 구성
  - VME-16 (1-연결기) : single height VME 보드 접속
    - \* Connector P1 (BD) J1 (Backplane) 사용
    - \* 16비트 데이터, 24비트 어드레스로 구성됨
  - VME-32 (2-연결기) : double height VME 보드 접속시 이용됨
    - \* Connector P2 (BD) J2 (Backplane) 사용
    - \* 32비트 데이터, 32비트 어드레스로 구성됨
  - 1,2 연결기는 96 DIN 연결

### VMEbus의 구조및 기능 (Cont'd)

Slide 8

- VMEbus의 특성
  - 8, 16, 32 비트의 데이터 전달 가능 (DS0#, DS1#, LWORD# 이용)
  - 16, 24, 32 비트의 주소 전달 가능 (AM5-AM0 이용)
  - 비동기 interlock 전송 방식
    - \* 시스템 클럭에 동기되지 않고 DS0#, DS1#, DTACK# 등 이용
    - \* 에러시 무한정 기다리는것을 방지하기 위한 버스 타임아웃기능 (BERR# 이용)
  - 멀티 마스터가 가능 (최대 20개까지 마스터가 존재하여도 됨)
  - 여러개의 인터럽트 핸들러가 가능

### VMEbus의 구성

- VME 버스의 신호 (J1, P1)

Slide 9

Table 1 J1/P1 Pin Assignments

PIN NUMBER	ROW a	ROW b	ROW c
	SIGNAL MEMEMONIC	SIGNAL MEMEMONIC	SIGNAL MEMEMONIC
1	D00	HRISV*	D08
2	D01	HC1L*	D09
3	D02	AC*FAIL*	D10
4	D03	BC0IN*	D11
5	D04	HC0OUT*	D12
6	D05	BC1IN*	D13
7	D06	HC1OUT*	D14
8	D07	BC2IN*	D15
9	GND	HC2OUT*	GND
10	SYSCLK	IC3IN*	SYS*FAIL*
11	GND	HC3OUT*	BERR*
12	DS1*	HR0*	SYS*RESET*
13	DS0*	BR1*	LWORD*
14	WRITE*	BR2*	AM5
15	GND	BR3*	A23
16	DTACK*	AM0	A22
17	GND	AM1	A21
18	AS*	AM2	A20
19	GND	AM3	A19
20	IACK*	SBUS_DATA	A18
21	IACKIN*	SBUS_STRB	A17
22	IACKOUT*	GND	A16
23	AM4	GND	A15
24	A07	HR7*	A14
25	A06	HR6*	A13
26	A05	HR5*	A12
27	A04	HR4*	A11
28	A03	HR3*	A10
29	A02	HR2*	A09
30	A01	HR1*	A08
31	-12V	+5VSTRIB	+12V
32	+5V	+5V	+5V

## VMEbus의 구성 (Cont'd)

- VME 버스의 신호 (J2, P2)

Table 2 J2/P2 Pin Assignments

ROW a PIN NUMBER	ROW b SIGNAL MEMMONIC	ROW c SIGNAL MEMMONIC	SIGNAL MEMMONIC
1	User Defined	+5V	User Defined
2	User Defined	GND	User Defined
3	User Defined	RETRY	User Defined
4	User Defined	A24	User Defined
5	User Defined	A25	User Defined
6	User Defined	A26	User Defined
7	User Defined	A27	User Defined
8	User Defined	A28	User Defined
9	User Defined	A29	User Defined
10	User Defined	A30	User Defined
11	User Defined	A31	User Defined
12	User Defined	GND	User Defined
13	User Defined	+5V	User Defined
14	User Defined	D16	User Defined
15	User Defined	D17	User Defined
16	User Defined	D18	User Defined
17	User Defined	D19	User Defined
18	User Defined	D20	User Defined
19	User Defined	D21	User Defined
20	User Defined	D22	User Defined
21	User Defined	D23	User Defined
22	User Defined	GND	User Defined
23	User Defined	D24	User Defined
24	User Defined	D25	User Defined
25	User Defined	D26	User Defined
26	User Defined	D27	User Defined
27	User Defined	D28	User Defined
28	User Defined	D29	User Defined
29	User Defined	D30	User Defined
30	User Defined	D31	User Defined
31	User Defined	GND	User Defined
32	User Defined	+5V	User Defined

Slide 10

## VMEbus의 구성 (Cont'd)

- 주소 신호 (A01-A31)
  - short address (A01-A15) → 64KB (16비트 주소)
  - standard address (A01-A23) → 16MB (24비트 주소)
  - extended address (A01-A31) → 4GB (32비트 주소)
  - AM5-AM0 (address modify) 이용하여 지정
- 데이터 신호 (D00-D31)
  - Multiplex 되지 않은 비동기 전송
  - 8, 16, 32 비트 데이터 전송 가능
  - low-end data justification 사용
  - DS0#, DS1#, LWORD# 을 이용하여 전송데이터 size 지정

Slide 11

## VMEbus의 구성 (Cont'd)

---

- Slide 12**
- 제어신호
    - AS# : 주소가 유효함을 알림
    - DS0# : 홀수바이트 데이터가 유효함을 알림 (D0-D7 유효)
    - DS1# : 짝수바이트 데이터가 유효함을 알림 (D8-D15 유효)
    - DTACK# : 해당 데이터에 대한 동작을 끝냈음을 표시
    - LWORD# : 32비트데이터가 유효함을 알림 (D0-D31 유효)
    - WRITE# : 읽기/쓰기 신호
    - BERR# : 오류가 발생하였을때 DTACK# 대신에 생성됨
      - \* 예) 무효주소 access
      - \* 예) 포트 넓이가 32비트가 아닌데 32비트로 액세스한경우등
    - IACK# : 인터럽트가 인지되었음을 알림

## VMEbus의 구성 (Cont'd)

---

- Slide 13**
- 유틸리티 그룹
    - SYSCLK : 16MHz 시스템 클럭 신호
    - SYSRESET\* : 시스템 리셋 신호
    - SYSFAIL\* : 시스템 다운 신호
    - ACFAIL\* : AC 전압 저하 신호 (신호 저하시 배터리 백업장치 사용)
  - 직렬통신용 신호
    - SERCLK, SERDAT

## VMEbus의 시스템 구성

---

### Slide 14

- 시스템 클럭 : 16MHz (SYSCLK) 공급
  - 비동기식 버스를 사용함으로 CPU 클럭과 꼭 같을 필요 없음
- 아비터 : 여러 DTB (Data Transfer Bus) 리퀘스터의 버스 사용을 조정
  - 반드시 슬롯 A01내에 존재하여야함
- controller sub-system (A01 내에 존재하여야 함)
  - DTB 아비터
  - 시스템 클럭 드라이버
  - 시스템 리셋 드라이버 (선택적, A01 이 아닌보드 가능)
  - AC fail 드라이버 (선택적, A01 이 아닌보드 가능)
  - 버스 타임아웃검출 모듈 (선택적)

## VMEbus의 시스템 구성 (Cont'd)

---

### Slide 15

- 마스터 : 데이터의 전송 사이클을 시작하는 기능
  - 어드레스 버스 구동
  - AM 코드, AS#, DS1#, DS0#, WRITE# 구동
  - DTB 마스터
  - 최소 하나, 최대 20개 가능
- Request : DTB 의 제어를 아비터에게 요구할 능력이 있는 모듈
- 인터럽트 핸들러 : 인터럽트에 대하여 응답할 능력을 갖는 모듈
  - 보통 마스터 및 Request 와 함께 마스터 서브시스템으로 구성됨
- 마스터 서브 시스템 : 마스터, Requestor, 인터럽트 핸들러로 구성됨



## VMEbus의 시스템 구성 (Cont'd)

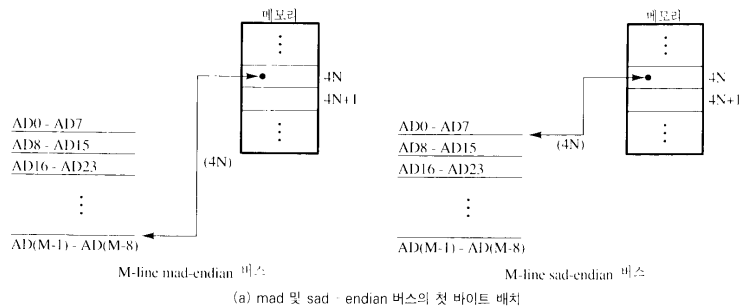
### Slide 16

- 슬레이브 : 마스터의 데이터 전송에 응답하는 모듈
  - DTACK# 및 BERR# 드라이브 능력이 있어야함
  - DTB 슬레이브로 불림
- 인터럽터 : 인터럽트 요구를 발생하는 모듈
- 슬레이브 서브 시스템: 슬레이브와 인터럽터의 기능을 가지는 시스템
  - 보통 슬레이브 기능만 갖는 메모리 시스템은 슬레이브라고 부름

## VMEbus의 접속

- MAD-Endian 및 SAD-Endian 시스템 버스와 접속
  - MAD-Endian : 낮은 주소의 데이터가 높은 데이터 바이트 (big-endian 과 정합)
  - SAD-Endian : 낮은 주소의 데이터가 낮은 데이터 바이트 (little-endian 과 정합)

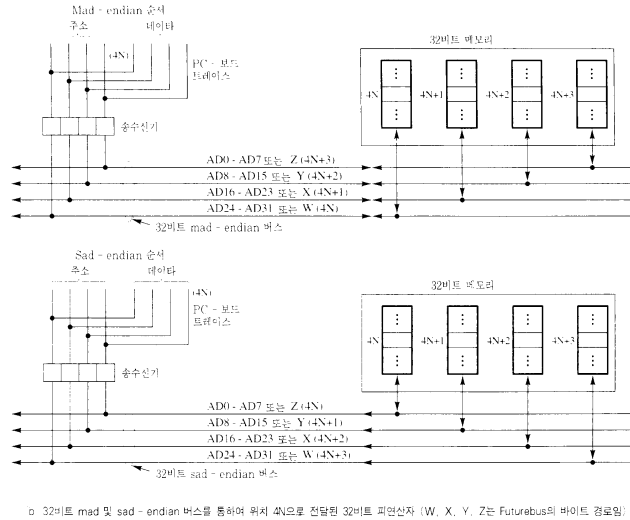
### Slide 17



### VMEbus의 접속 (Cont'd)

- MAD-Endian 과 SAD-Endian 의 데이터 전달

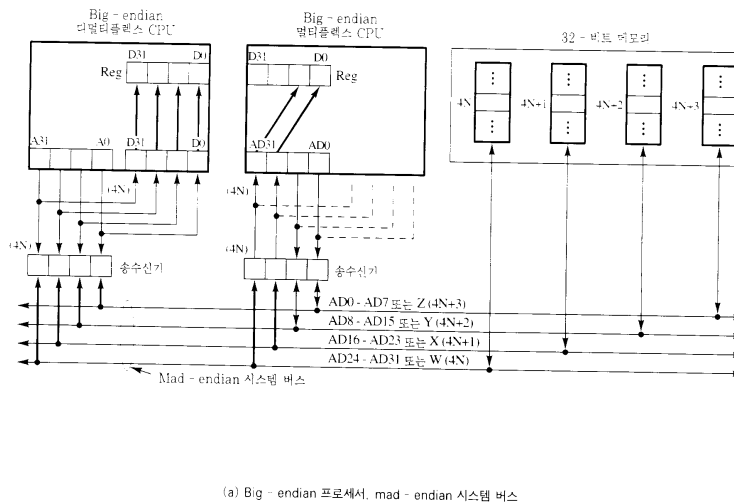
Slide 18



### VMEbus의 접속 (Cont'd)

- Big-Endian 프로세서, mad-endian 시스템 버스

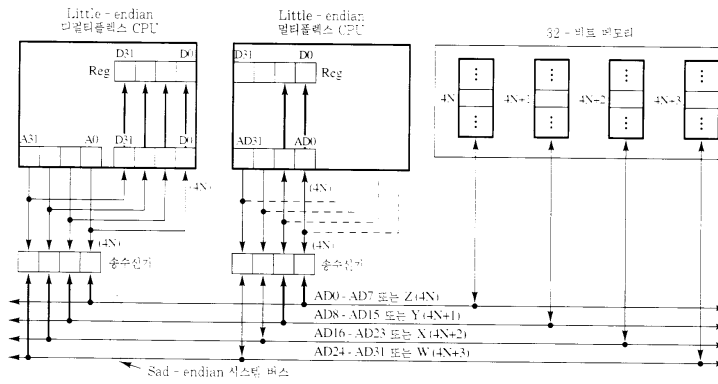
Slide 19



### VMEbus의 접속 (Cont'd)

- little-Endian 프로세서, sad-endian 시스템 버스

Slide 20

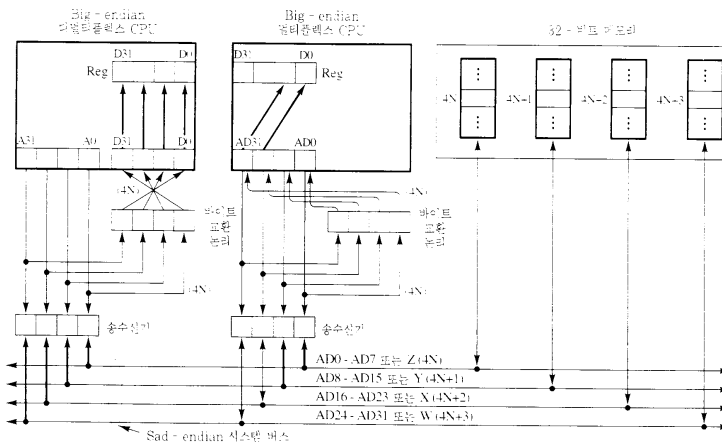


(b) Little-endian 프로세서, sad-endian 시스템 버스

### VMEbus의 접속 (Cont'd)

- 정합되지 않는 마이크로 프로세서와 시스템 버스

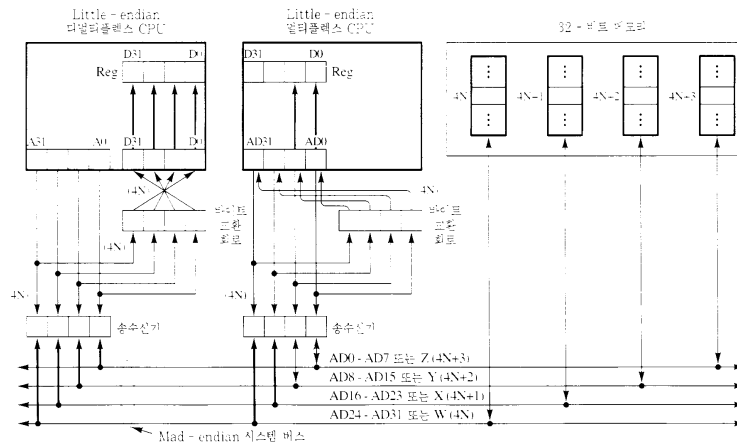
Slide 21



a) Big-endian 프로세서 sad-endian 시스템 버스

## VMEbus의 접속 (Cont'd)

Slide 22



b) Little-endian 프로세서, mad-endian 시스템 버스

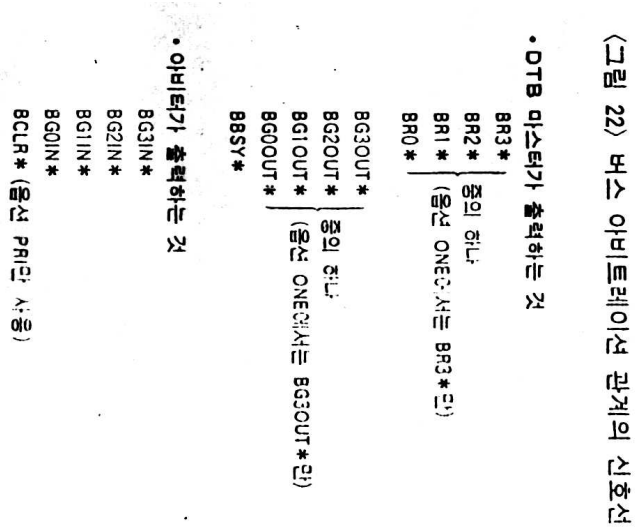
## VME Arbitration Interface

Slide 23

- DTB 아비트레이션 신호들
  - BR3# BR0# : 버스 요구선들
  - BG3IN# BG0IN#, BG3OUT# BG0OUT# : 버스 아비트레이션 데이저 체인 신호
  - BBSY#, BCLR# : 버스의 상태를 나타내는 신호선

### VME Arbitration Interface (Cont'd)

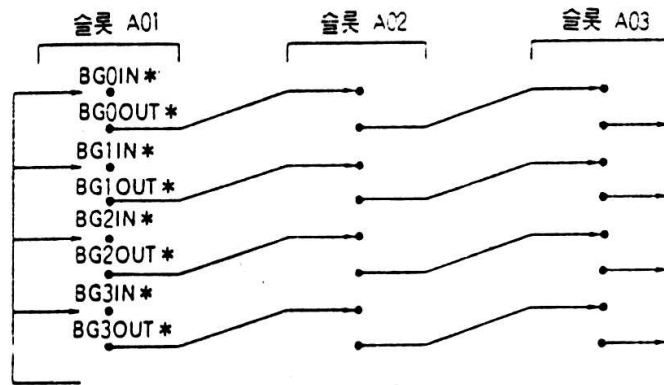
- 버스 아비트레이션 관계의 신호선



Slide 24

### VME Arbitration Interface (Cont'd)

- 버스 grant daisy chain



(주) 슬롯 A01의 버스 그랜트 인라인은 슬롯 A01에 끼워진 보드상에 통상 위치하는 아비터에 의해 드라이브 된다.

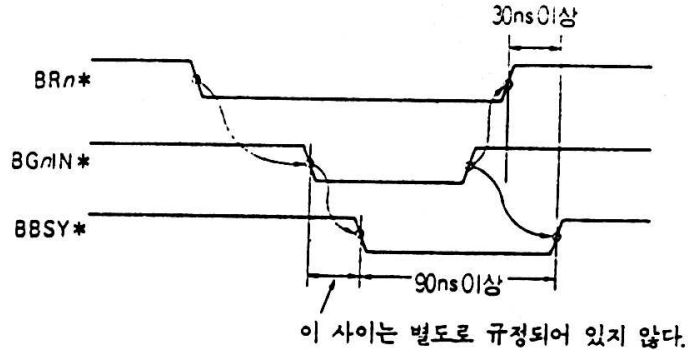
Slide 25

### VME Arbitration Interface (Cont'd)

- 버스 아비트레이션 DTB 마스터측 타이밍

〈그림 23〉 버스 아비트레이션 DTB 마스터측 타이밍

Slide 26

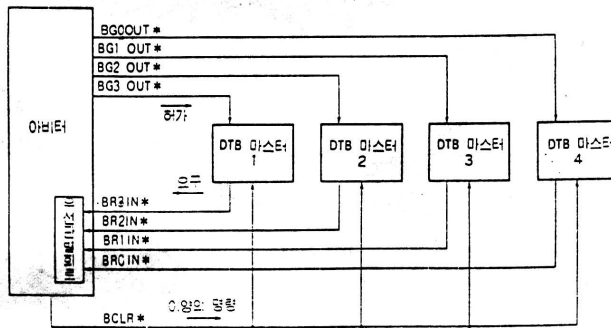


### VME Arbitration Interface (Cont'd)

- PRI 방식

〈그림 18〉 옵션 PRI (고정 우선방식)

Slide 27



이비터는 현재 버스 사용중인 DTB 마스터보다 우선도가 높은 DTB 마스터에서의 BR이 있으면  $BCLR^*$ 을 출력하여 버스 사용권을 높은 측에 넘겨준다.

## VME Arbitration Interface (Cont'd)

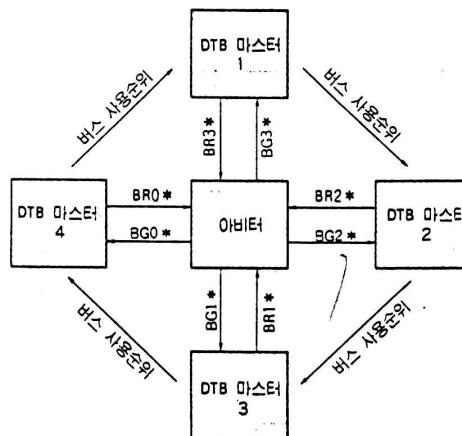
### Slide 28

- BR3IN\* BR0IN\* 및 BG3OUT\* BG0OUT\* 의 우선순위 사용
- 우선 순위가 높은 Request 가 들어오면 BCLR\* 로 버스의 이양을 요구
- 버스를 놓으면 (AS\* 로 체크) 높은 버스마스터에게 버스 이양
- DMA 는 높은 우선순위를 주어야함
- I/O 보드나 CPU 보드는 낮은순위 가능

## VME Arbitration Interface (Cont'd)

- RRS 방식

### Slide 29



아비터터는 현재 버스 사용중인 DTB 마스터에서 오른쪽 방향으로 차례로 버스 사용권을 부여한다. 예를 들면, DTB 마스터 2 사용중에 1에서 요구가 있더라도 버스 사용권을 그 다음인 3에 부여하여 3은 다시 4로, 4는 다시 1로 부여한다.

## VME Arbitration Interface (Cont'd)

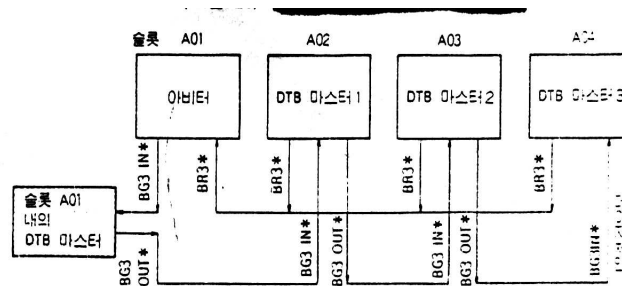
Slide 30

- BR3IN\* BR0IN\* 및 BG3OUT\* BG0OUT\* 의 우선순위 사용
- 순서대로 우선순위가 결정됨
- 가장 늦게 사용한 매스터 가장 낮은 우선순위
- 해당 순위의 버스 매스터가 버스 요구가 없으면 일정시간 기다린후 다음으로 넘어감
- 모든 매스터가 CPU 보드일 때 사용가능
- BCLR\* 은 사용되지 않음 →RWD 와 ROR 옵션 가능
- RWD (Release With Done) 버스 사용후 곧바로 버스를 놓음
- ROR (Release On Request) 다른 매스터의 요구가 있으면 놓음

## VME Arbitration Interface (Cont'd)

- ONE 방식

Slide 31



각 DTB 매스터는 자신이 BR3\*를 어서트하고 있지 않은 경우, BG3 IN\*의 어서트를 감지하면 BG3 OUT\* 어서트하여 다음 슬롯의 보드로 전송한다



## VME Arbitration Interface (Cont'd)

---

- Slide 32**
- BR3\* 만사용
  - 우선순위에는 BG3IN\* BG3OUT\* 의 데이터 체인 이용
  - 슬롯의 위치에 의해 우선순위가 결정됨

## VME Interrupt

---

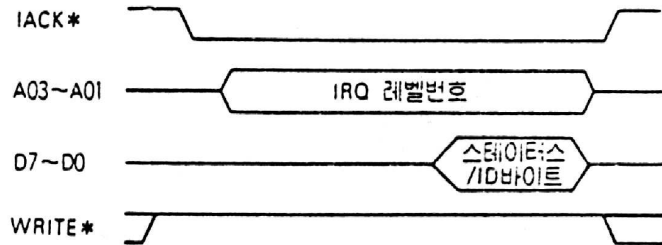
- Slide 33**
- 우선인터럽트 그룹
    - 인터럽트 요구선 (IRQ7# IRQ1#)
    - 인터럽트 응답선 (IACK#) : 인터럽트 ack. 시 A03 A01 IRQ 레벨  
→D7 D0 는 상태/ ID 바이트 가 들어감
    - 인터럽트 응답 데이터 체인용 신호 (IACKIN#, IACKOUT#)

## VME Interrupt (Cont'd)

- 인터럽트 과정

〈그림 1〉 IACK\* 어서트시의 A03~A01, D07~D00의 사용방법

Slide 34



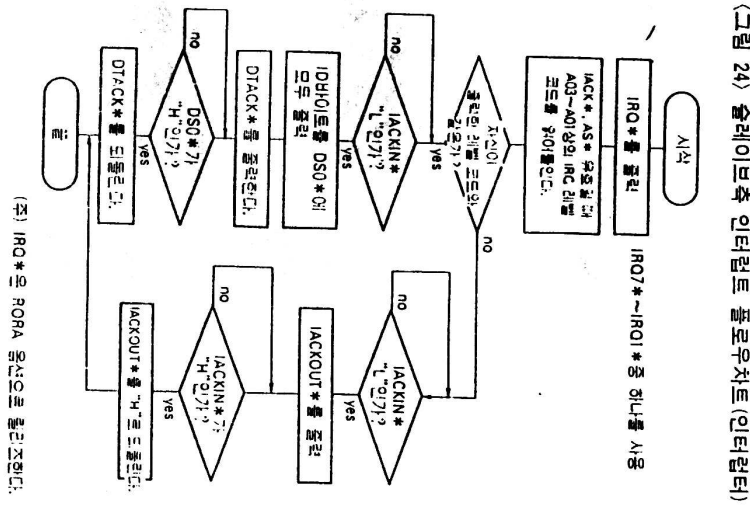
## VME Interrupt (Cont'd)

Slide 35

- 슬레이브 서브 시스템 : IRQ7\* IRQ1\* 중 하나를 assert
- 인터럽트 핸들러 : IACK\* 를 assert , IACKIN\* 과 IACKOUT\* 를 통하여 신호전달
- 인터럽트 핸들러 : A03 A01 에 인터럽트 레벨을 줌
- IACKIN\* 받은 서브 시스템 자신의 요구 IRQ 레벨과 비교
  - \* 일치하면
    - 데이터버스상에 (D07 D00) 스태이터스/ID 바이트 송출, DTACK\*
    - IACKOUT\* 송출안함
  - \* 일치하지 않으면
    - 바로 IACKOUT\* 송출

### VME Interrupt (Cont'd)

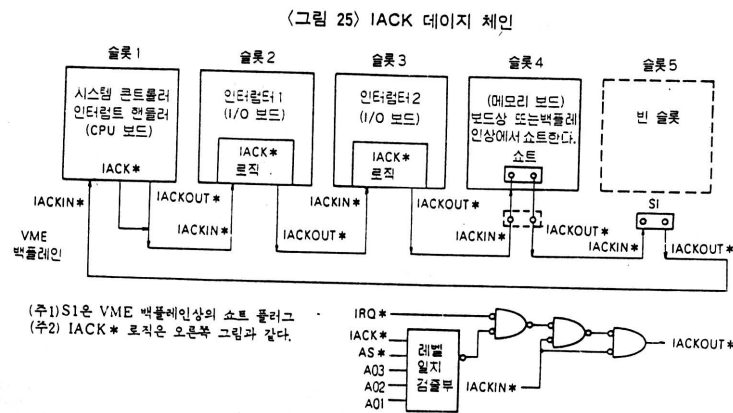
- 슬레이브 측 인터럽트 flowchart



Slide 36

### VME Interrupt (Cont'd)

- 데이지 체인을 제공하지 않는 보드는 백플레인상에 IACKIN\* 과 IACKOUT\*를 연결



Slide 37

## VME Interrupt (Cont'd)

- IRQ 레벨 표

VME  
A01

〈표 4〉 IRQ 레벨 코드 일람표

핸들러에 의해 검지된 IRQ * 번호	핸들러에 의해 출력된 IRQ 레벨 번호		
	A03	A02	A01
IRQ7 *	H	H	H
IRQ6 *	H	H	L
IRQ5 *	H	L	H
IRQ4 *	H	L	L
IRQ3 *	L	H	H
IRQ2 *	L	H	L
IRQ1 *	L	L	H

Slide 38

〈그림 26〉 스테이터스/ID 바이트 송출 타이밍