

# 시스템 버스 (system bus)

- ◆ 시스템버스
  - ◆ 시스템버스의 조직
  - ◆ 시스템버스의 동작
- ◆ 버스중재 방식
  - ◆ 병렬중재방식
  - ◆ 직렬중재방식
  - ◆ 폴링방식

이 자료는 김종현저-컴퓨터구조론(생능출판사)의 내용을 편집한 것입니다.

# 7.1 시스템 버스 (system bus)

- 컴퓨터시스템의 구성 요소들(CPU, 기억장치, I/O 장치들)을 상호 연결해주는 중심 통로

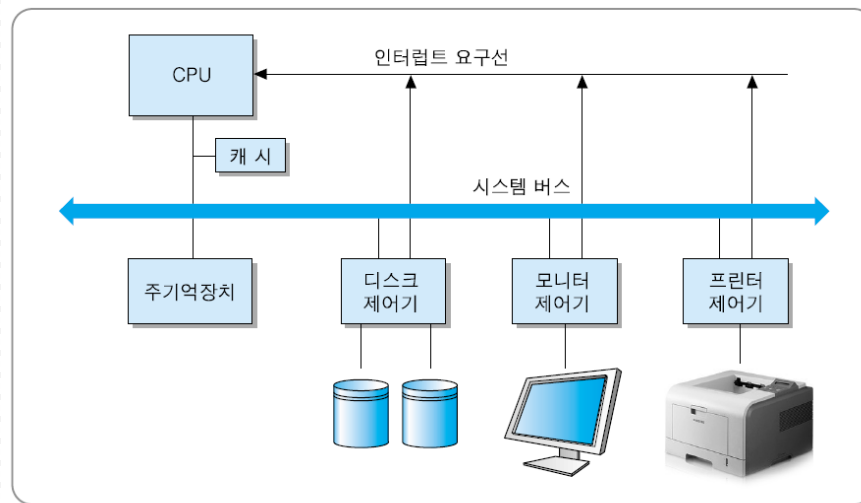


그림 7-1 시스템 버스를 이용한 컴퓨터시스템의 구성도

## 7.1.1 시스템 버스의 조직

- **버스(bus)** : 구성 요소들간에 교환할 각종 정보들을 전송하는 선(line) 들로 구성
- **버스 선의 수**
  - ▣ 한 번에 전송하는 데이터 비트들의 수, 기억장치 주소 비트들의 수 및 제어 신호들의 수에 따라 결정됨
  - ▣ 소형 컴퓨터 : 50 ~ 100 개의 신호선들 사용
  - ▣ 중대형급 시스템 : 100 개 이상의 신호선들 사용

# 기능에 따른 버스의 종류

## □ 데이터 버스(data bus)

- 시스템 요소들 사이에 데이터를 전송하는 데 사용되는 선들의 집합
- 양방향 전송(bidirectional transfer)
- 버스 폭(선들의 수) = CPU와 기억장치 사이에 한 번에 전송되는 비트 수

## □ 주소 버스(address bus)

- CPU가 기억장치로 (또는 기억장치로부터) 데이터 쓰기 (또는 읽기) 동작을 할 때, 해당 기억장소를 지정하는 주소를 전송하기 위한 선들의 집합
- 단방향 전송(unidirectional transfer) : CPU → 기억장치 및 I/O 제어기
- 주소 버스의 비트 수에 의해 시스템에 접속될 수 있는 전체 기억장치 용량이 결정됨
- 직접 주소지정 할 수 있는 기억장소의 단위 : 바이트(byte) 혹은 단어(word)

## □ 제어 버스(control bus)

- CPU와 기억장치 및 I/O 장치 사이에 제어 신호들을 전송하는 선들의 집합

## 주소 버스의 폭(비트 수)에 따른 기억장치 용량 예

- 주소 버스 = 16비트 인 경우
  - ▣ 주소지정 가능한 최대 기억장소들의 수 =  $2^{16} = 65,536(64K)$  개
  - ▣ 바이트 단위 주소 지정일 경우 최대 기억장치 용량 = 64 Kbyte
- 주소 버스 = 24비트 인 경우
  - ▣ 최대  $2^{24} = 16,777,216(16M)$  개
  - ▣ 바이트 단위 주소 지정일 경우 최대 기억장치 용량 = 16 Mbyte

# 제어 버스

- 기억장치 및 I/O 장치와의 데이터 교환을 위한 제어신호들
  - 기억장치 쓰기(memory write) 신호 : 버스에 실린 데이터를 주소가 지정하는 기억장소에 저장되도록 하는 제어 신호
  - 기억장치 읽기(memory read) 신호 : 주소가 지정하는 기억장소의 내용을 읽어서 버스에 실리게 하는 제어 신호
  - I/O 쓰기(I/O write) 신호 : 버스에 실린 데이터를 지정된 I/O 장치로 출력되게 하는 제어 신호
  - I/O 읽기(I/O read) 신호 : 지정된 I/O 장치로부터 데이터를 읽어서 데이터 버스에 실리게 하는 제어 신호
- 버스 중재를 위한 제어신호들(이 신호들의 집합을 중재 버스라고도 부름)
- 인터럽트 메커니즘을 위한 제어신호들(이 신호들의 집합을 인터럽트 버스라고도 부름)

## 중재 버스(arbitration bus)

- **버스 마스터(bus master)**: 시스템 버스에 접속되는 요소들 중에서 버스 사용의 주체가 되는 요소들 (예: CPU, 기억장치 모듈, I/O 제어기, 등)
- **버스 중재 (bus arbitration)**: 시스템 버스에 접속된 두 개 또는 그 이상의 버스 마스터들이 동시에 버스를 사용하고자 할 때 순서대로 한 개의 마스터씩 버스를 사용할 수 있게 해주는 동작
- **중재 버스**: 버스 중재를 위한 신호 선들의 집합
  - ▣ **버스 요구(bus request) 신호**: 버스 마스터가 버스 사용을 요구했음을 알리는 신호
  - ▣ **버스 승인(bus grant) 신호**: 버스 사용을 요구한 마스터에게 사용을 허가하는 신호
  - ▣ **버스 사용중(bus busy) 신호**: 현재 버스가 사용되고 있는 중임을 나타내는 신호

# 인터럽트 버스(interrupt bus)

- 인터럽트 메커니즘을 위한 제어 신호선들의 집합
  - ▣ 인터럽트 요구(interrupt request) 신호 : I/O 장치가 인터럽트를 요구했음을 알리는 신호
  - ▣ 인터럽트 확인(interrupt acknowledge) 신호 : CPU가 인터럽트 요구를 인식했음을 알리는 신호
- 그 이외의 제어신호들
  - ▣ 버스 클럭(bus clock) 신호 : 동기식 버스에서 버스 동작들의 시작 시간을 일치시키기 위하여 제공되는 공통 클럭 신호
  - ▣ 리셋(reset) 신호 : 모든 시스템 요소들의 동작을 초기화시키는 신호



# 버스 대역폭(bus bandwidth)

- 버스의 속도를 나타내는 척도로서, 단위 시간당 전송할 수 있는 데이터 양을 나타내며, 버스 클럭의 주기에 의해 결정

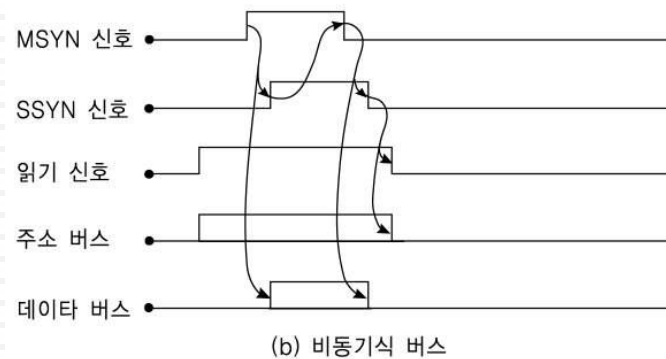
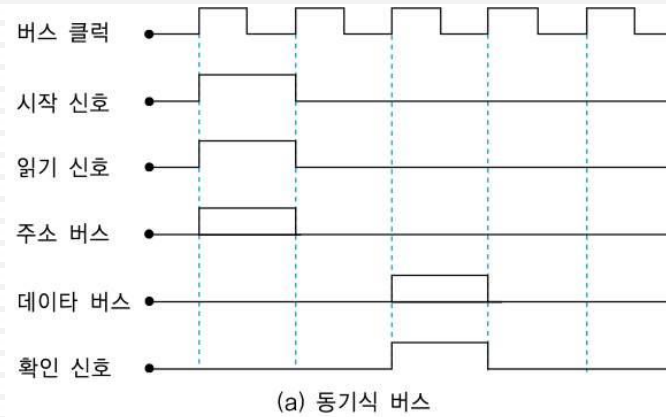
[예] 버스 클럭의 주기:  $50\text{ ns}$  (클럭 주파수:  $20\text{ MHz}$ )  
데이터 버스의 폭: 64 비트(8 바이트)

→ 버스 대역폭 =  $8\text{ byte} / (50 \times 10^{-9}\text{ sec}) = 160\text{ [Mbytes/sec]}$   
(즉, 이 버스를 통하여 초당 1억 6천 바이트의 데이터 전송 가능)

## 7.1.2 시스템 버스의 기본 동작

- 쓰기 동작(write operation) 순서
  - ① 버스 마스터가 버스 사용권 획득
  - ② 버스를 통하여 주소와 데이터 및 쓰기 신호 전송
- 읽기 동작(read operation) 순서
  - ① 버스 마스터가 버스 사용권 획득
  - ② 주소와 읽기 신호를 보내고, 데이터가 전송되어 올 때까지 대기
- 버스 동작의 타이밍에 따른 버스의 분류
  - 동기식 버스(synchronous bus) : 시스템 버스에서 모든 버스 동작들이 공통의 버스 클럭을 기준으로 발생
  - 비동기식 버스(asynchronous bus) : 버스 동작들의 발생 시간이 관련된 다른 버스 동작의 발생 여부에 따라 결정

# 동기식 버스와 비동기식 버스의 시간 흐름도



# 동기식 / 비동기식 버스의 비교

## □ 동기식 버스

[장점] 인터페이스 회로 간단

[단점] 버스 클럭의 주기가 가장 오래 걸리는 버스 동작의 소요 시간을 기준으로 결정되므로, 클럭 주기보다 더 짧은 시간이 걸리는 버스 동작의 경우에는 동작이 완료된 후에도 다음 주기가 시작될 때까지 대기

## □ 비동기식 버스

[장점] 각 버스 동작이 완료 즉시 연관된 다음 동작이 발생하기 때문에 동기식 버스에서와 같이 낭비되는 시간이 없음

[단점] 연속적 동작을 처리하기 위한 인터페이스 회로가 복잡

## □ 소규모 컴퓨터 : 비동기식 버스 사용

일반적인 컴퓨터시스템 : 동기식 버스 사용

## 7.2 버스 중재 (bus arbitration)

- **버스 경합(bus contention)** : 한 개의 시스템 버스에 접속된 여러 개의 버스 마스터들이 동시에 버스 사용을 요구하는 현상
- **버스 중재(bus arbitration)** : 버스 경합이 발생하는 경우, 어떤 기준에 따라 버스 마스터들 중에서 한 개씩만 선택하여 순서대로 버스를 사용할 수 있게 해주는 동작
- **버스 중재기(bus arbiter)** : 버스를 중재하는 하드웨어 모듈

# 버스 중재 방식의 분류

## □ 제어 신호들의 연결 구조에 따른 중재 방식의 분류

### □ 병렬 중재 방식(parallel arbitration scheme)

- 각 버스 마스터들이 독립적인 버스 요구 신호를 발생하고, 버스 승인 신호를 받음  
→ 버스 마스터들의 수와 같은 개수의 버스 요구 선 및 승인 신호 선 필요

### □ 직렬 중재 방식(serial arbitration scheme)

- 버스 요구와 승인 신호 선이 각각 한 개씩만 존재하며, 각 신호 선을 버스 마스터들 간에 직렬로 접속하는 방식

# 버스 중재 방식의 분류

## □ 버스 중재기의 위치에 따른 분류

### ▣ 중앙집중식 중재 방식(centralized arbitration scheme)

- 시스템 내에 버스 중재기가 한 개만 존재하는 방식
- 버스 마스터들이 발생하는 버스 요구 신호들은 하나의 중재기로 보내지고, 중재기는 정해진 중재 원칙에 따라 선택한 버스마스터에게 승인 신호를 발생

### ▣ 분산식 중재 방식(decentralized arbitration scheme)

- 여러 개의 버스 중재기들이 존재하며(일반적으로 각 버스 마스터가 중재기를 한 개씩 가짐), 버스 중재 동작이 각 마스터의 중재기에 의하여 이루어지는 방식

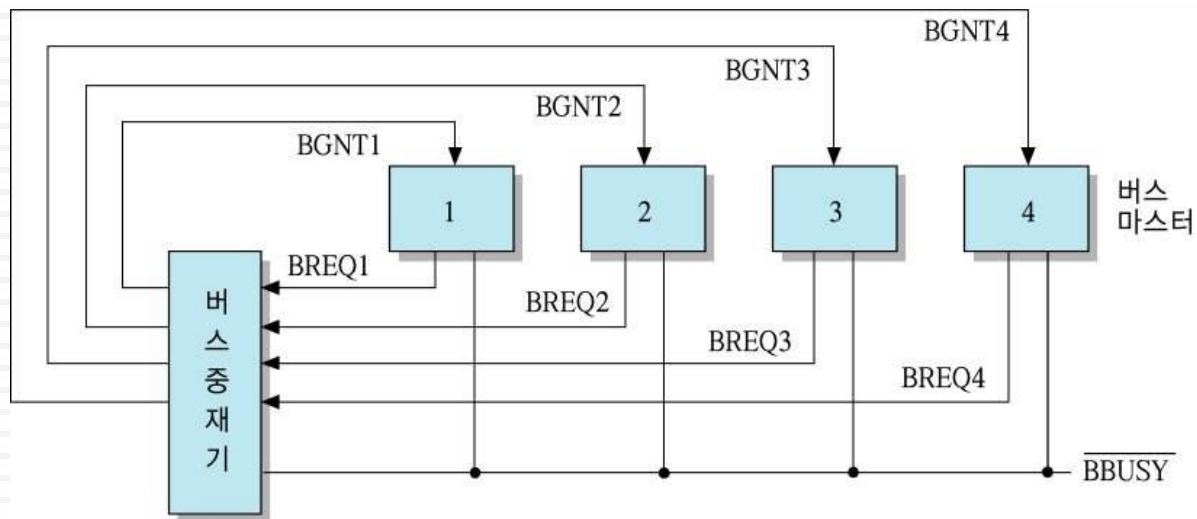
## 7.2.1 병렬 중재 방식

- 우선순위의 결정 방식에 따른 분류
  - ▣ 고정-우선순위 방식(fixed-priority scheme) : 각 버스 마스터에 지정된 우선순위가 고정되어 있는 방식
  - ▣ 가변-우선순위 방식(dynamic-priority scheme) : 우선순위를 변경할 수 있는 방식



# 1) 중앙집중식 고정-우선순위 중재방식

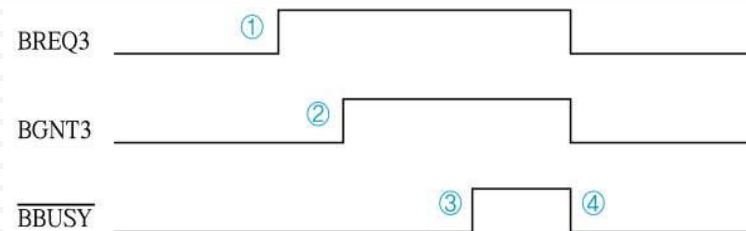
- 모든 버스 마스터들이 버스 중재기에 접속
- 중재기와 가장 가까이 위치한 버스 마스터 1이 가장 높은 우선 순위, 버스 마스터 4가 가장 낮은 우선순위를 가지는 것으로 가정



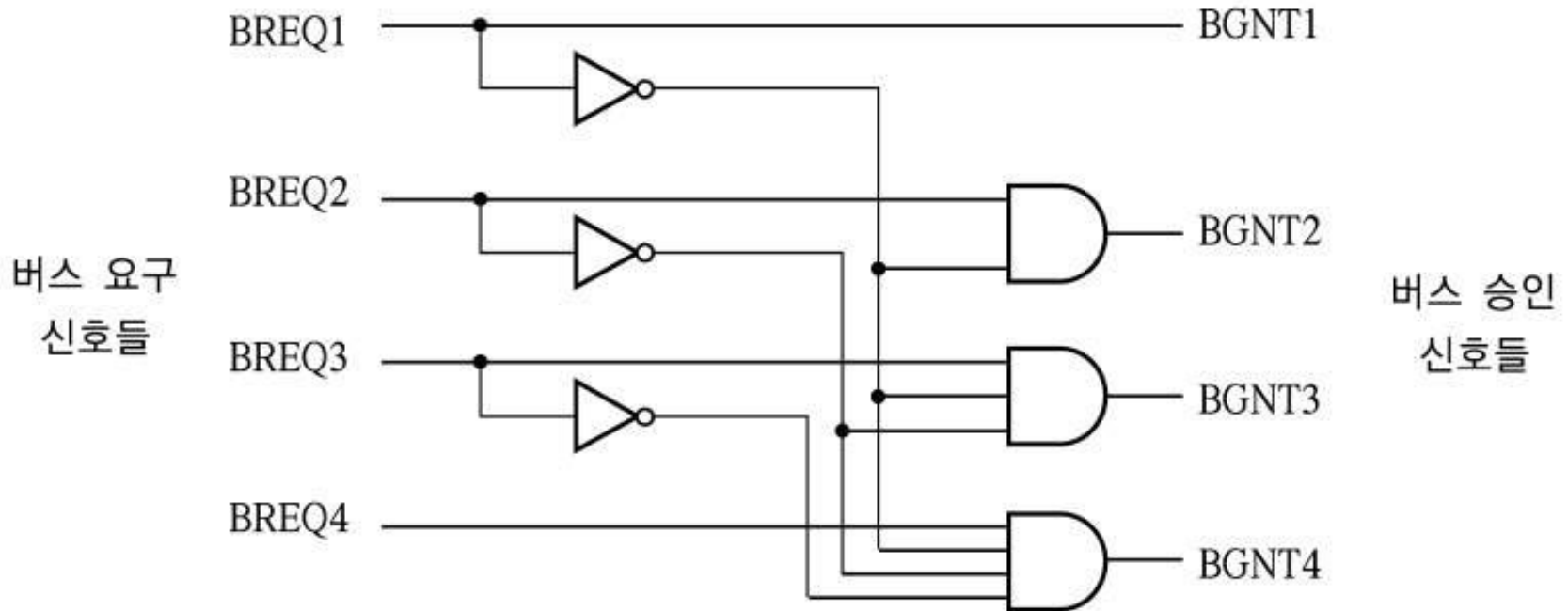
# 중앙집중식 고정 우선순위 중재 방식 (계속)

[예] 버스마스터 1이 버스를 사용중일 때, 버스마스터 3이 버스를 요구

1. 마스터 3이 **BREQ3** 신호를 세트
2. 버스 중재기가 마스터 3에게 **BGNT3** 신호를 세트하여 버스를 허가
3. 마스터 1이 버스를 사용을 끝내고 **BBUSY** 신호를 해제
4. 마스터 3이 **BBUSY** 신호를 다시 세트하고 버스를 사용 시작.  
이때 **BREQ3**와 **BGNT3**는 제거됨



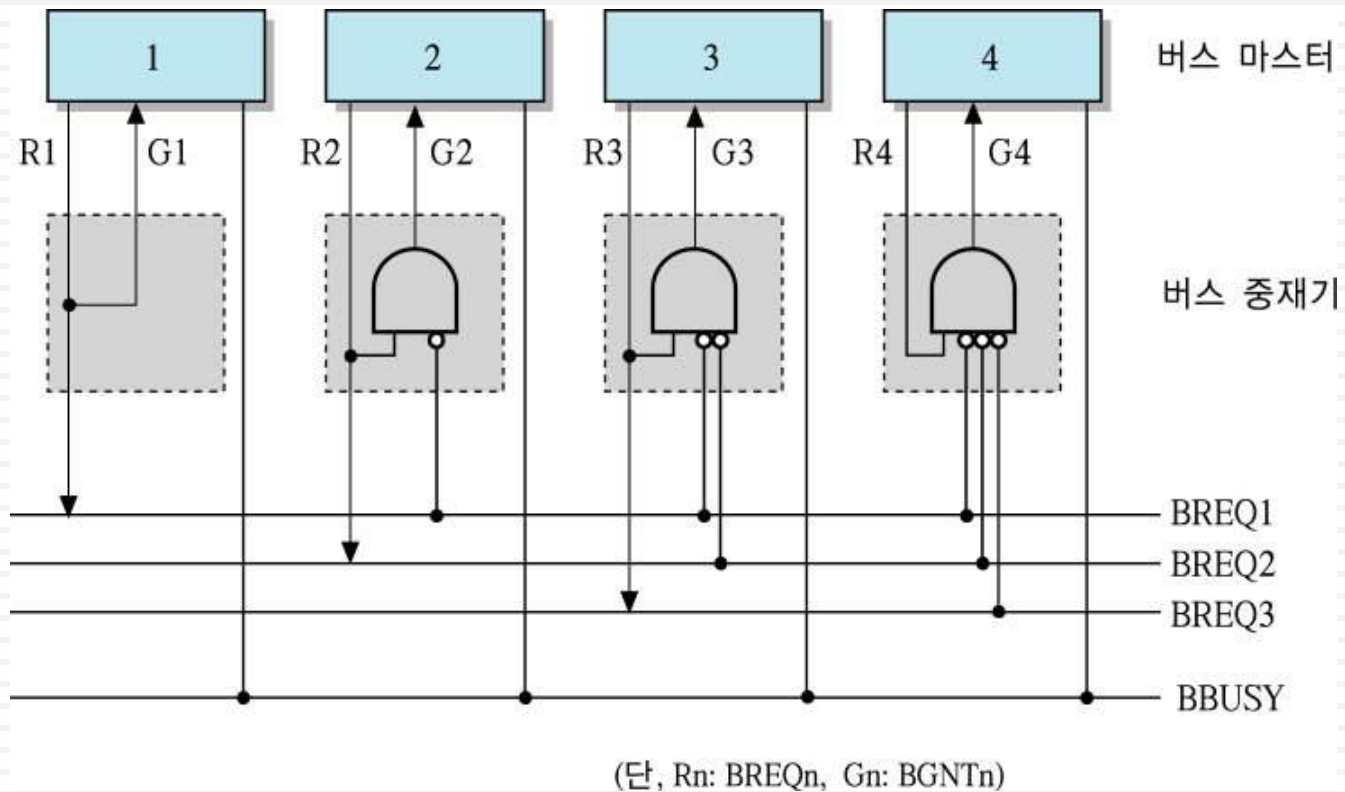
# 병렬 중재기의 내부 회로도



## 2) 분산식 고정-우선순위 방식

- 모든 버스 마스터들이 중재기를 한 개씩 보유
- 중재 동작
  - ▣ 각 중재기는 자신보다 더 높은 우선순위를 가진 마스터들의 버스 요구 신호들을 받아서 검사하여, 그들이 버스 사용 요구를 하지 않은 경우에만 자신의 버스 마스터로 버스 승인 신호 발생
  - ▣ 승인 신호를 받은 버스 마스터는 **BBUSY** 신호를 검사하여, 비활성화 상태(다른 마스터가 버스를 사용하지 않는 상태)일 때 버스 사용을 시작
- 분산식 중재 방식의 장단점
  - [장점] 중앙집중식에 비하여 중재 회로가 간단하므로 동작 속도가 빨라진다
  - [단점] 고장을 일으킨 중재기를 찾아내는 방법이 복잡하고, 한 중재기의 고장이 전체 시스템의 동작에 영향을 미칠 수가 있다

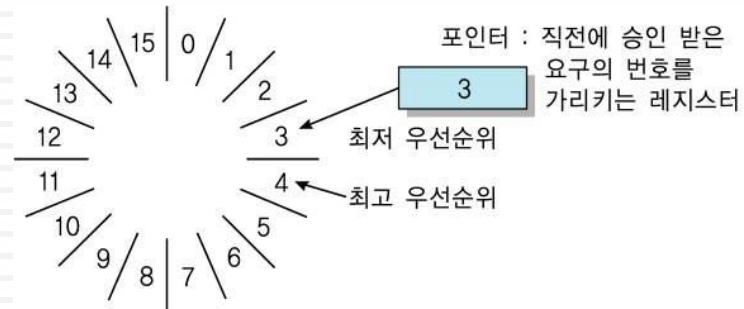
# 분산식 고정-우선순위 방식의 구성도



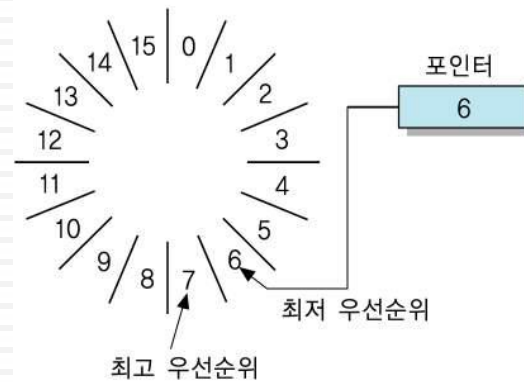
### 3) 가변 우선순위 방식

- 시스템의 상태(또는 조건)에 따라 각 버스 마스터들의 우선순위를 계속 변화시키는 방식
  - [단점] 중재 회로 복잡
  - [장점] 모든 마스터들이 공정하게 버스를 사용할 수 있게 해준다
- 회전 우선순위(rotating priority) 방식
  - [방법1] 중재 동작이 끝날 때마다 모든 마스터들의 우선순위가 한 단계씩 낮아지고, 가장 우선순위가 낮았던 마스터가 최상위 우선순위를 가지도록 하는 방법
  - [방법2] 일단 버스 사용 승인을 받은 마스터는 최하위 우선순위를 가지며, 바로 다음에 위치한 마스터가 최상위 우선순위를 가지도록 하는 방법  
→ Acceptance-dependent식 회전 우선순위 방식

# Acceptance-dependent식 회전 우선순위 방식의 예



(a) 마스터 3의 요구가 승인된 후



(b) 마스터 6의 요구가 승인된 후

## 가변 우선순위 방식 (계속)

- 임의 우선순위 방식 : 각 중재 동작이 끝날 때마다 우선 순위를 임의로 결정.
- 동등 우선순위 방식 : 모든 마스터들이 동등한 우선순위를 가지며, FIFO(First-In First-Out) 알고리즘 사용
- 최소-최근 사용(Least-Recently Used: LRU) 방식 : 최근 가장 오랫동안 버스를 사용하지 않은 버스 마스터에게 최상위 우선순위 할당  
[단점] 회로가 매우 복잡



## 7.2.2 직렬 중재 방식

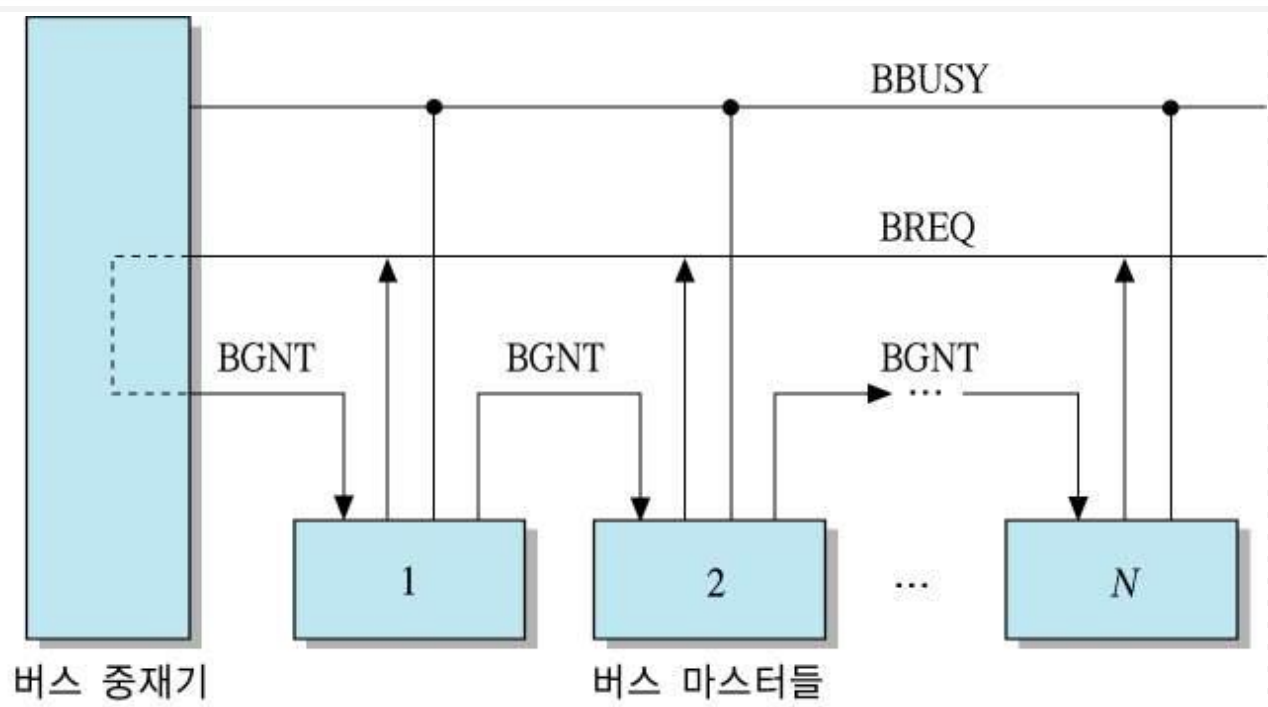
### 1) 중앙집중식 직렬 중재 방식

- ▣ 하나의 중재 신호선(BGNT)이 데이지-체인(daisy-chain)형태로 모든 버스 마스터들을 직렬로 연결
- ▣ 우선순위는 버스 승인 신호선이 연결된 순서대로 결정
- ▣ 동작 원리
  - ▣ 한 개 또는 그 이상의 버스 마스터가 버스 사용을 요구하면 공통의 BREQ 신호가 세트
  - ▣ 버스 중재기 : 데이지 체인의 첫 번째에 접속된 마스터로 승인 신호(BGNT) 전송
  - ▣ BGNT 신호를 받은 마스터는 만약 버스 사용을 요구한 상태라면, 버스 사용권을 가짐

## 중앙집중식 직렬 중재 방식 (계속)

- 만약 버스 사용을 요구하지 않은 상태라면, 승인 신호를 다음에 연결된 마스터로 통과
- 승인 신호는 버스를 요구한 마스터에게 도달할 때까지 계속 통과  
→ 버스 요구를 보낸 마스터들 중에서 중재기에 가장 가까이 위치한 (우선순위가 가장 높은) 마스터에게 승인 신호가 전달되면 그 마스터가 버스 사용권을 획득

# 중앙집중식 직렬 중재 방식의 구성도 (데이터 체인)



## 2) 분산식 직렬 중재 방식

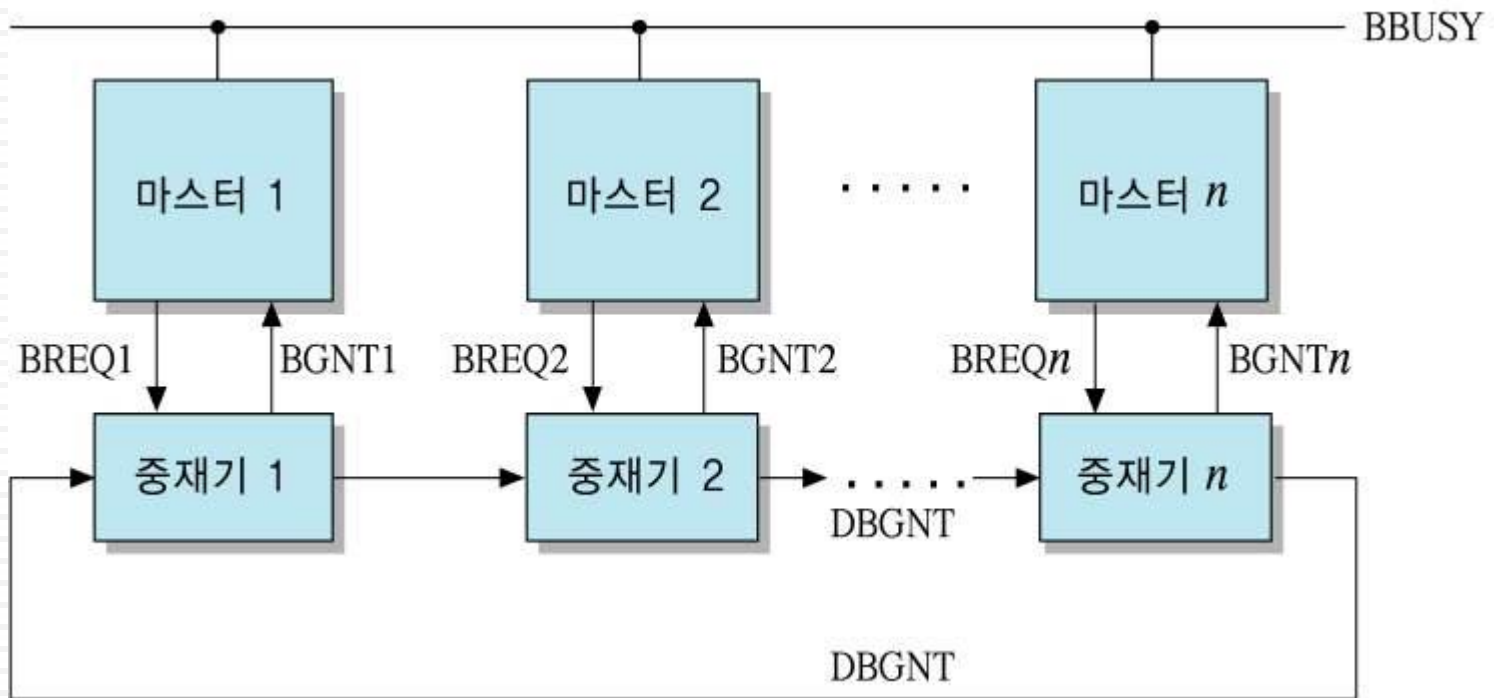
### □ 구성

- 데이지-체인 버스 승인 신호(daisy-chained bus grant signal: DBGNT)가 버스 중재기들을 순환형(circular)으로 접속

### □ 동작 원리

- 버스 사용권을 부여 받은 마스터가 버스 사용을 시작하는 순간에 (그 마스터의 중재기는) 자신의 우측에 위치한 마스터의 중재기로 접속된 **DBGNT** 신호를 세트
- 만약 그 마스터가 버스 사용을 신청하고 기다리던 중이었다면, 중재기는 즉시 **DBGNT** 신호를 받아들여서 **BGNT** 신호를 발생시켜 마스터로 전송
- **DBGNT** 신호를 받은 마스터가 버스 요구를 하지 않은 상태라면, 그 신호를 우측의 다음 중재기로 통과시키고, 그러한 과정은 버스를 요구한 마스터에 도달할 때까지 반복

# 분산식 직렬 중재 방식의 구성도



## 분산식 직렬 중재 방식 (계속)

- **특징** : 각 마스터의 우선순위가 계속 변화
  - ▣ 버스 사용 승인을 받으면 다음 중재 동작에서는 최하위 우선순위를 가짐
  - ▣ 버스를 사용한 마스터의 바로 우측에 위치한 마스터가 최상위 우선순위를 가짐
  - ▣ 순환형 구조에서 **DBGNT** 신호가 연결된 순서대로 우선순위가 하나씩 감소
- **단점** : 어느 한 지점에만 결함이 발생해도 전체 시스템의 동작 중단

## 7.2.3 폴링 방식

### □ 폴링 방식(polling scheme)의 원리

- ▣ 버스 사용을 원하는 마스터가 있는지를 버스 중재기가 주기적으로 검사하여 사용 승인 여부를 결정

### 1) 하드웨어 폴링 방식

- ▣ 버스 중재기와 각 버스 마스터 간에 별도의 폴링 선(polling line)이 존재
- ▣ 2진 코드화된 폴링 주소(binary encoded polling address)를 이용하면, 폴링 선의 수가  $\log_2 N$  개로 감소
- ▣ 공통의 **BREQ** 선과 **BBUSY** 선이 각각 한 개씩 존재

## 하드웨어 폴링 방식 (계속)

### □ 동작 순서

- 중재기는 폴링 주소를 발생하여 검사할 마스터를 지정한 다음에, 그 마스터가 버스 사용을 원하는지 묻는다
- 지정된 마스터가 버스 사용을 원하면 **BREQ** 신호를 세트
- **BREQ** 신호가 세트되면, 중재기는 현재 검사 중인 마스터에게 버스 사용을 허가하고, 그렇지 않으면(지정된 마스터가 버스 사용을 원하지 않으면) 다음 마스터들에 대한 검사를 순서대로 진행

### □ 우선순위 결정 방법

- 중재기가 마스터를 검사하는 순서에 의하여 결정되며, 검사할 마스터의 번호는 2진 카운터(binary counter)를 이용하여 발생



# 하드웨어 폴링 방식의 구성도

