

컴퓨터 구조

1

시스템 버스 (제 12주 차)

서울사이버대학교

오 창 환

학습 목표

2

- 시스템 버스를 설명할 수 있다.
- 버스 중재를 설명할 수 있다.

학습 내용

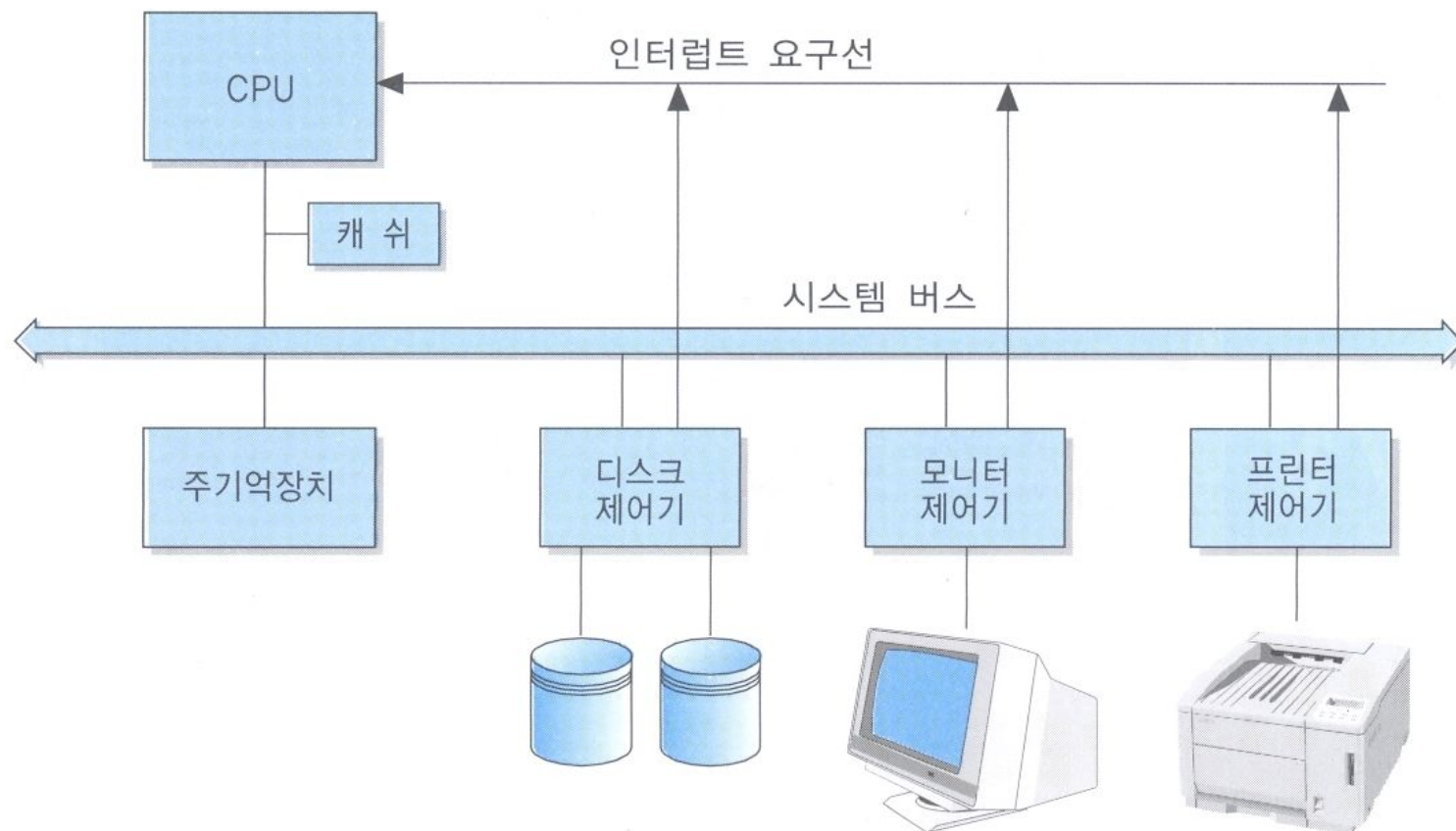
3

- 시스템 버스
- 버스 중재

시스템 버스 (1)

4

- 컴퓨터시스템을 구성하는 주요 요소인 CPU, 주기억장치 및 입출력장치(Input/Output device : I/O 장치)들은 시스템 버스를 통하여 서로 접속됨.



- 시스템 버스를 이용한 컴퓨터시스템의 구성도

시스템 버스 (2)

5

(1) 시스템 버스의 조직

- 시스템 버스에 접속되는 모든 요소들은 버스를 통하여 상호간에 데이터를 전송하고, 동작 시간을 조정하기 위한 정보도 교환함.
- 따라서 버스는 많은 수의 전기 도체 선들로 구성되는데, 일반적으로 소형 컴퓨터에서는 50개에서 100개 정도이고 중대형 급 시스템에서는 그 이상임.
- 이 선들은 기능에 따라 다음과 같이 크게 세 가지 버스들로 나누어지며, 각 버스는 전송되는 정보의 종류를 나타내는 이름을 가지고 있음.
 - * 데이터 버스(data bus)
 - * 주소 버스(address bus)
 - * 제어 버스(control bus)

시스템 버스 (3)

6

- 데이터 버스는 시스템 요소들 사이에 데이터를 전송하는 데 사용되는 선들의 집합임.
CPU가 기억장치로부터 한 번에 8 비트씩 읽어 온다면 필요한 데이터 버스 선의 수는 8개가 되며, 32 비트씩 읽어오는 시스템에서는 32개로 구성됨.

데이터는 CPU와 기억장치, CPU와 입출력장치, 기억장치와 입출력장치 사이에 양방향으로 전송되기 때문에 버스 인터페이스 회로도 양방향 전송을 지원할 수 있어야 함.

- 주소 버스는 CPU가 기억장치로 (또는 기억장치로부터) 데이터 쓰기 (또는 읽기) 동작을 할 때, 해당 기억장소를 지정하기 위한 주소를 전송하는 선들의 집합임.

CPU가 I/O장치를 사용할 때도 주소가 필요하므로 각 I/O 장치로도 주소 버스가 연결되어야 하며 주소는 CPU에 의해서만 발생되기 때문에 단방향 전송 기능만 있으면 됨.

주소 버스 선의 수는 CPU가 주소지정 할 수 있는 전체 기억장치 용량을 결정해 줌.

예를 들면, 16 비트 주소 버스로 주소를 지정할 수 있는 기억장소의 수는 $2^{16} = 65,536$ (즉 64K)개, 24비트인 경우에는 $2^{24} = 16,777,216$ (즉, 16M)개가 됨.

시스템 버스 (4)

7

- 제어 버스를 구성하는 제어 신호 선들은 각각 고유의 기능을 가지고 있으므로, 시스템의 구성과 동작에 따라 제어 신호의 종류와 수도 달라지게 됨.
 - * 먼저 CPU가 기억장치 및 I/O 장치와 데이터를 교환하는 데 필요한 제어 신호들은 다음과 같음.
 - 기억장치 쓰기(memory write) 신호 : 버스 상의 데이터가 지정된 기억장소에 쓰여지게(저장되게) 함.
 - 기억장치 읽기(memory read) 신호 : 지정된 기억장소의 내용을 읽어서 버스에 실리게 함.
 - 입출력 쓰기(I/O write) 신호 : 버스 상의 데이터가 지정된 I/O 장치로 출력되게 함.
 - 입출력 읽기(I/O read) 신호 : 지정된 I/O 장치로부터 데이터를 읽어서 데이터 버스에 실리게 함.
 - 전송 확인(transfer acknowledge) 신호 : 데이터 전송 동작이 완료되었음을 알려줌.

시스템 버스 (5)

8

* 시스템 버스에 접속되는 요소들 중에서 버스 사용의 주체가 되는 요소들을 버스 마스터(bus master)라고 하는데, 일반적으로 CPU와 I/O 제어기 등이 버스 마스터가 됨.

그러나 어느 한 순간에 한 개의 버스 마스터만 시스템 버스를 사용할 수 있기 때문에, 두 개 또는 그 이상의 마스터들이 동시에 버스를 사용하고자 할 때는 순서대로 사용하도록 중재(arbitration)를 해 주어야 하는데 이 동작을 위하여 다음과 같은 제어 신호들이 필요함.

- 버스 요구(bus request) 신호 : 어떤 마스터가 버스 사용을 원하고 있다는 것을 나타냄.
- 버스 승인(bus grant) 신호 : 버스 사용을 요구한 마스터에게 사용을 허가한다는 것을 나타냄.
- 버스 사용 중(bus busy) : 현재 어떤 마스터가 버스를 사용하고 있는 중이라는 것을 나타냄.

시스템 버스 (6)

9

- * CPU와 I/O 장치 간의 비동기적 동작을 지원하는 인터럽트 메커니즘을 위한 제어신호들은 다음과 같음.
 - 인터럽트 요구(interrupt request) 신호 : 어떤 I/O 장치가 인터럽트를 요구하고 있다는 것을 나타냄.
 - 인터럽트 확인(interrupt acknowledge) 신호 : CPU가 인터럽트 요구를 인식했다는 것을 나타냄.
- * 상기 이외의 제어 신호들은 다음과 같음.
 - 버스 클럭(bus clock) : 동기식 버스에서 버스 동작들의 시작 시간을 일치시키기 위하여 제공되는 공통 클럭 신호임.
 - 리셋(reset) : 모든 시스템 요소들의 동작을 초기화시키는 신호임.

시스템 버스 (7)

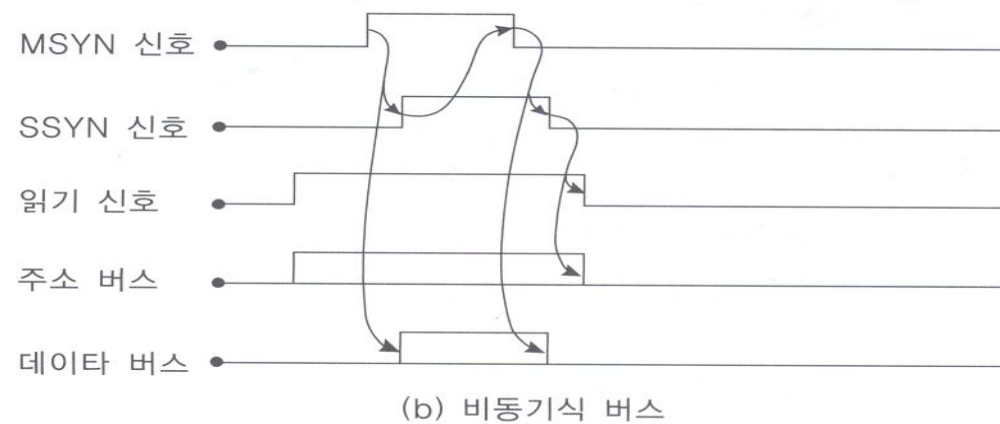
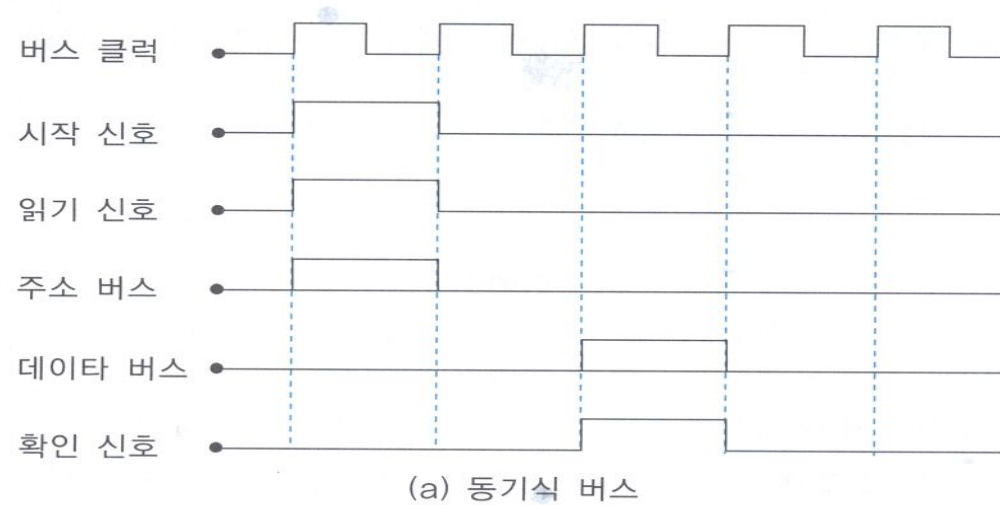
10

(2) 시스템 버스의 기본 동작

- 시스템 버스의 쓰기 동작(write operation) 순서는 다음과 같음.
 - ① 버스 마스터가 버스 사용권을 획득함.
 - ② 버스를 통하여 주소와 데이터 및 쓰기 신호를 보냄.
- 시스템 버스의 읽기 동작(read operation) 순서는 다음과 같음.
 - ① 버스 마스터가 버스 사용권을 획득함.
 - ② 주소와 읽기 신호를 보내고, 데이터가 전송되어 올 때까지 기다림.
- 시스템 버스에서 모든 버스 동작들이 발생하는 시간이 공통의 버스 클럭을 기준으로 결정되는 버스를 동기식 버스(synchronous bus)라고 부르고, 그와는 달리 버스 동작들의 발생 시간이 관련된 다른 버스 동작의 발생 여부에 따라 결정되는 비동기식 버스(asynchronous bus)가 있음.

시스템 버스 (8)

11



• 기억장치 읽기 동작의 시간 흐름도

시스템 버스 (9)

12

- 동기식 버스에서 기억장치로부터 데이터를 읽어오는 과정은 먼저, 버스 클럭의 첫 번째 주기 동안에 CPU가 주소와 읽기 제어 신호를 기억장치로 보내는데, 이때 버스 동작의 시작을 알리는 신호가 함께 보내지기도 함.

두 번째 주기 동안에 기억장치에서 데이터 인출 동작이 일어나며,

세 번째 주기에서 인출된 데이터를 기억장치가 버스를 통해 CPU로 전송함.

이와 함께 기억장치는 전송 확인 신호를 보내며

이 신호를 받은 CPU는 자신이 요청한 데이터가 버스에 실려있다는 것을 알고 받아들이게 됨.

동기식 버스는 인터페이스 회로가 간단하다는 장점이 있지만, 버스 클럭의 주기가

가장 오래 걸리는 버스 동작의 소요 시간을 기준으로 하여 정해져야 하기 때문에 ,

클럭 주기보다 더 짧은 시간이 걸리는 버스 동작의 경우에는 완료된 후에도 다음 주기가 시작될 때까지 기다려야 하는 단점이 있음.

시스템 버스 (10)

13

- 비동기식 버스의 읽기 동작에서는 CPU가 버스 상에 주소와 읽기 신호를 싣고, 신호가 안정될 때까지 잠시 기다린 후에, 주소와 제어 신호가 보내졌음을 알리기 위해 마스터 동기 신호(master synch signal: MSYN)를 보냄. 그러면 기억장치는 주소를 받아서 데이터를 인출한 다음에, 데이터를 버스에 싣고 슬레이브 동기 신호(slave synch signal: SSYN)를 보냄으로써 응답하게 됨. CPU는 SSYN 신호를 받은 즉시 데이터를 버스로부터 받아들이고, 주소와 읽기 신호 및 MSYN 신호를 제거함. MSYN 신호가 제거되는 것은 CPU가 데이터를 받아들였음을 나타내므로 기억장치도 SSYN 신호를 제거하며, 이것으로 읽기 동작이 완료됨. 비동기식 버스에서는 각 버스 동작이 완료되는 즉시 연관된 다음 동작이 일어나므로 동기식 버스에서와 같이 낭비되는 시간이 없지만, 이러한 연속적 동작을 처리하기 위한 인터페이스 회로가 복잡해지는 단점이 있음.

2 교시

버스 중재 (1)

15

- 한 개의 시스템 버스에 접속된 여러 개의 버스 마스터들이 동시에 버스 사용을 요구하는 경우에는 경쟁이 발생하게 되는데 이러한 현상을 버스 경합(bus contention)이라고 함.
- 버스 경합이 발생한 경우에 어떤 기준에 따라 버스 마스터들 중에서 한 개씩만 선택하여 순서대로 버스를 사용할 수 있게 해주는 동작을 버스 중재(bus arbitration)라 하며, 이러한 기능을 수행하는 하드웨어 모듈을 버스 중재기(bus arbiter)라 함.
- 버스 중재 방식은 제어 신호들(버스 요구 신호 및 버스 승인 신호)의 연결 구조에 따라 다음과 같이 두 가지로 분류할 수 있음.
 - * 병렬 중재 방식 : 각 버스 마스터가 독립적인 버스 요구 신호선을 가지고 있으며, 이들이 모두 중재 회로로 입력되고 버스 승인 신호도 각 버스 마스터에 대해 별도로 발생됨. 따라서 버스 마스터들의 수와 같은 수의 버스 요구 선 및 승인 신호 선들이 각각 필요하며, 이들이 모두 버스 중재기로 연결됨.
 - * 직렬 중재 방식 : 버스 요구와 승인 신호 선이 각각 한 개씩만 있으며, 각 신호 선이 버스 마스터들 간에 직렬로 접속됨. 이 경우에 접속되는 순서는 마스터들의 우선순위(priority)에 따라 결정됨.

버스 중재 (2)

16

- 버스 중재 방식은 버스 중재기의 위치에 따라 다음과 같이 분류할 수 있음.
 - * 중앙집중식 중재 방식 : 시스템 내에 버스 중재기가 한 개만 존재하며,
모든 버스 중재 기능은 이 중재기에 의해 이루어짐.
즉 버스 마스터들이 발생하는 버스 요구 신호들은 하나의 중재기로 보내지고,
중재기는 정해진 중재 원칙에 따라 선택한 버스 마스터에게 승인 신호를 보내게 됨.
 - * 분산식 중재 방식 : 여러 개의 버스 중재기들이 존재함.
일반적으로 각 버스 마스터가 중재기를 한 개씩 가짐.
따라서 버스 중재 동작이 각 마스터의 중재기에 의해 이루어짐.

버스 중재 (3)

17

(1) 병렬 중재 방식

(가) 중앙집중식 고정-우선순위 방식

- 아래 그림은 버스 마스터가 네 개인 경우에 중앙집중식 고정-우선순위 중재 방식의 회로 구성도를 보여주고 있는데, 이 구성에서 중재기에 가까이 위치한 순서대로 우선 순위가 정해졌다고 가정한다면, 버스 마스터 1이 가장 높은 우선순위를 가지며, 버스 마스터 4가 가장 낮은 우선순위를 가지게 됨.

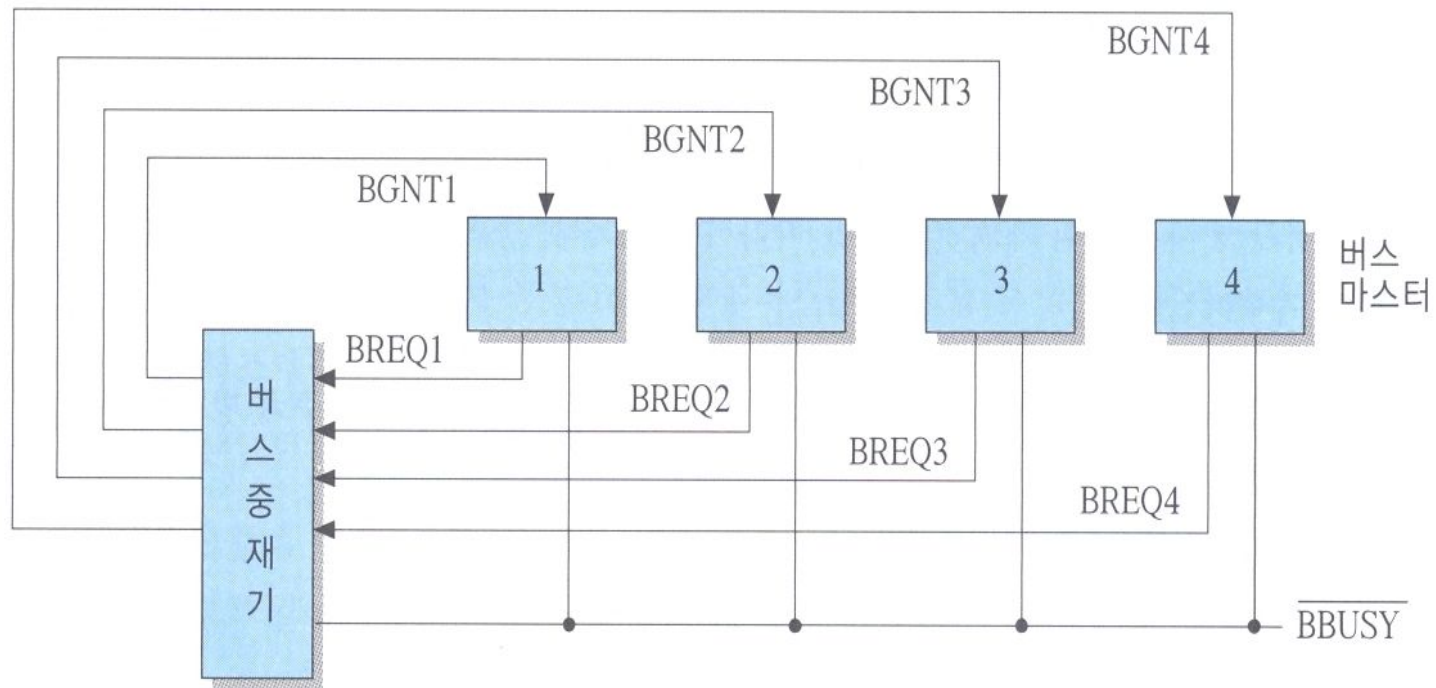
이 방식에서는 각 버스 마스터가 자신의 버스 요구(BREQ) 선을 가지고 있으며, 이들은 모두 하나의 버스 중재기로 접속됨.

버스 중재기는 한 개 또는 그 이상의 버스 요구 신호를 받아서 정해진 중재 원칙에 따라 버스 승인(BGNT) 신호들 중의 한 개만 세트 함.

BBUSY 신호는 어떤 버스 마스터가 버스를 사용하고 있는 중임을 나타내므로, 버스 승인을 받은 마스터는 그 신호가 세트 되어 있는 동안에는 기다렸다가, 해제되는 순간부터 버스 사용을 시작하게 됨. 또한 그 마스터는 버스 사용을 시작하는 순간에 BBUSY 신호를 세트 하여 다른 마스터가 버스를 사용하지 못하게 막아야 함.

버스 중재 (4)

18



- 중앙집중식 고정 우선순위 중재 방식의 구성도

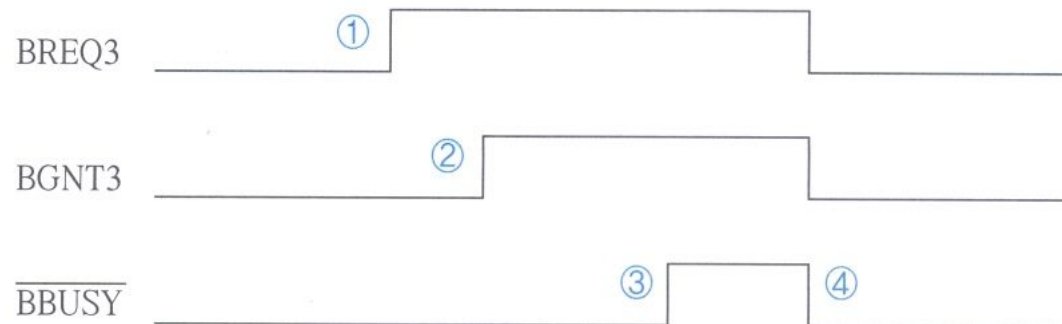
버스 중재 (5)

19

- 아래 시간 흐름도(timing diagram)는 버스가 사용되고 있는 중의 어느 한 순간에 대한 제어 신호들의 관계를 보여주고 있음.

예를 들어, 버스 마스터 1이 현재 버스를 사용하고 있는 중일 때 버스 마스터 3이 버스 사용을 요구한 경우의 중재 과정을 보면 다음과 같음.

- ① 마스터 3이 BREQ3 신호를 세트 함.
- ② 버스 중재기가 마스터 3에게 BGNT3 신호를 세트 하여 버스사용을 허가함.
- ③ 마스터 1이 버스 사용을 끝내고 BBUSY 신호를 해제함.
- ④ 마스터 3이 BBUSY 신호를 다시 세트 하고 버스 사용을 시작함. 이때 BREQ3와 BGNT3는 제거 됨.



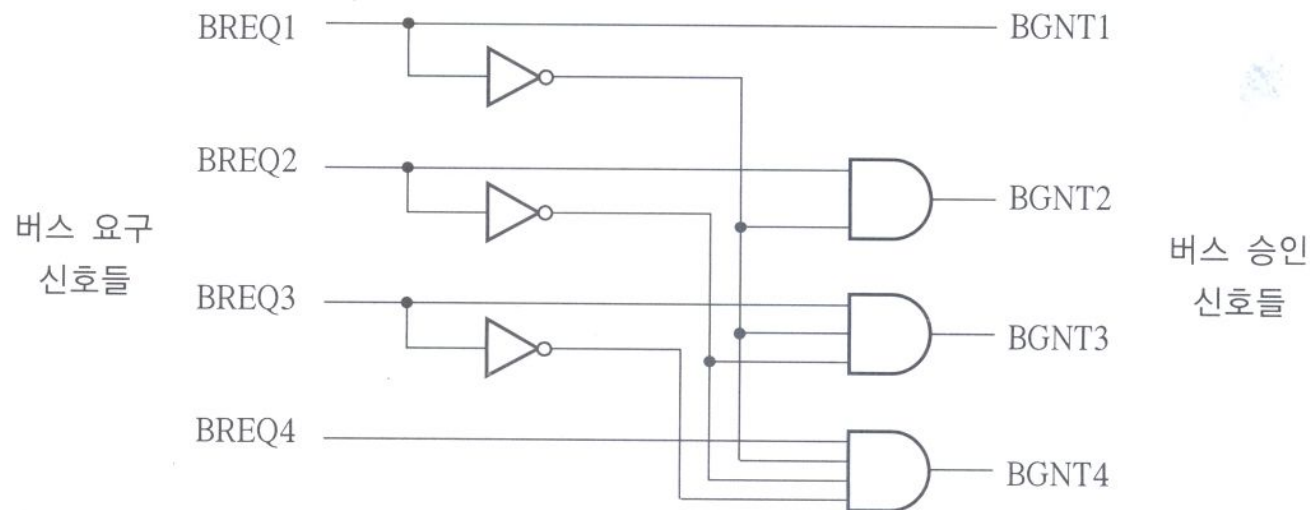
버스 중재 (6)

20

- 이 예에서 만약 버스 마스터 3이 버스 요구를 보내기 전에 우선순위가 더 낮은 버스 마스터 4가 버스를 사용하고 있었다면, 마스터 4에 의한 버스의 사용은 도중에 중단되지 않음.

버스 마스터 2와 3이 동시에 버스 요구를 보냈다면 우선순위가 더 높은 버스 마스터 2가 선택되어 BGNT2가 세트 됨.

- 각 마스터에 대한 BGNT 신호는 더 높은 우선순위를 가진 마스터가 버스 요구를 보내지 않은 상태에서 BREQ를 보냈을 때만 활성화 될 수 있음.



버스 중재 (7)

21

(나) 분산식 고정-우선순위 방식

- 이 방식에서는 모든 버스 마스터들이 중재기를 한 개씩 가지고 있음.

각 버스 마스터의 중재기는 자신보다 더 높은 우선순위를 가진 모든 버스 마스터들의 버스 요구 신호들을 입력으로 받아서 검사하고, 그들이 버스 사용 요구를 하지 않은 경우에만 자신의 버스 마스터로 버스 승인 신호를 보내줌.

- 분산식 중재 방식은 중앙집중식에 비해 중재 회로가 간단하기 때문에 동작 속도가 더 빠르고 신뢰도가 높은 장점이 있음.

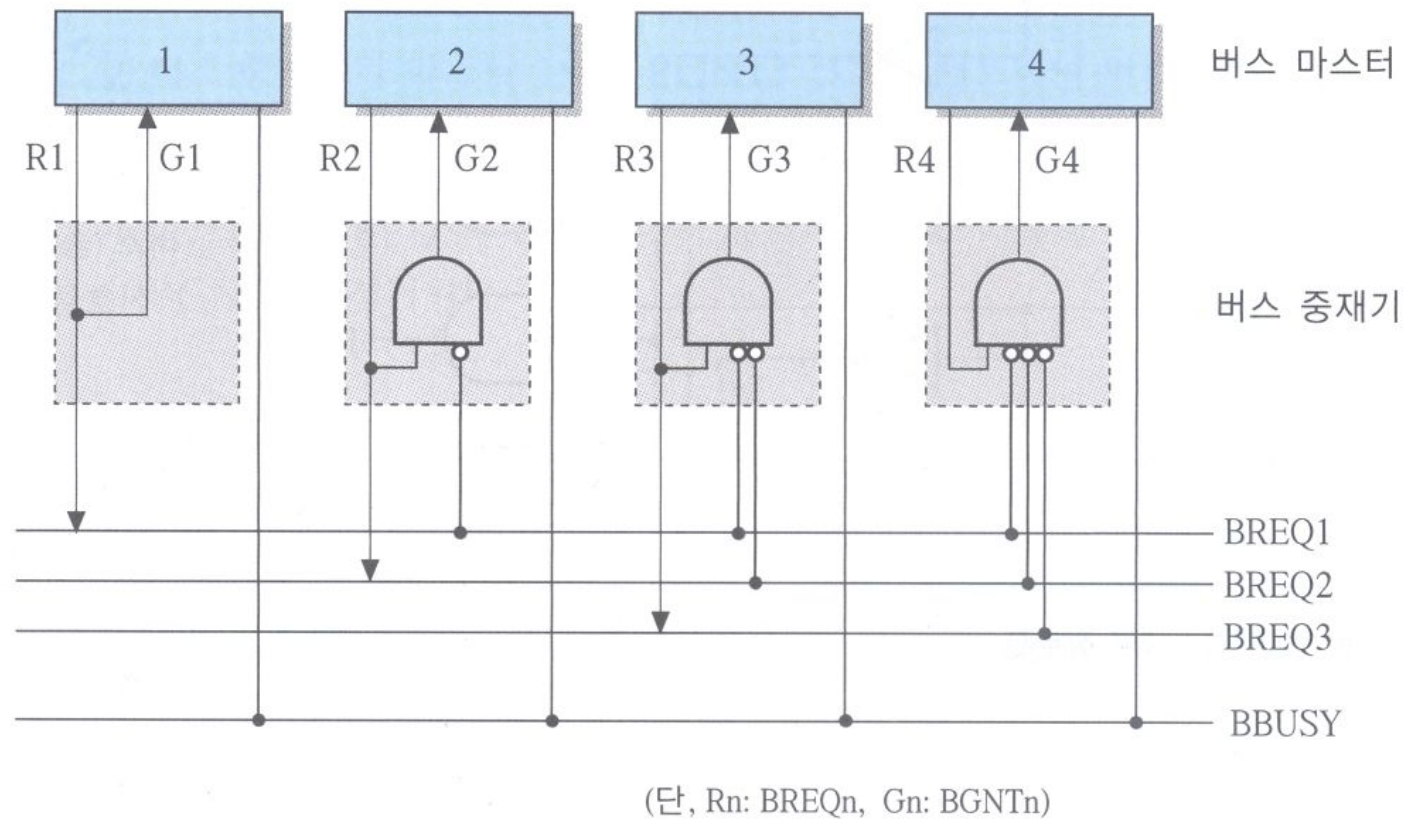
신뢰도가 높아지는 이유는 어떤 중재기가 고장이 나더라도 해당 마스터에만 영향을 미치지 때문임.

그러나, 고장을 일으킨 중재기를 찾아내는 방법이 복잡해지는 점과, 한 중재기의 고장이 전체 시스템의 동작에 영향을 줄 수도 있다는 문제가 있음.

예를 들면, 고장 난 중재기가 버스 승인 신호를 잘못 발생시키는 경우에는 두 개의 마스터들이 동시에 버스를 사용하게 될 수도 있음.

버스 중재 (8)

22



- 분산식 고정 우선순위 중재 방식의 구성도

버스 중재 (9)

23

(다) 가변 우선순위 방식

- 이 방식에서는 버스 중재기가 시스템의 상태에 따라 각 버스 마스터들의 우선순위를 계속 바꾸어 줌.

이 방식을 사용하면 중재 회로는 더 복잡해지지만,

버스 마스터들이 버스를 좀더 균등하게 사용할 수 있도록 해줄 수 있음.

즉, 우선순위가 고정되어 있는 방식에서 최상위 우선순위를 가진 마스터가 버스를 독점하거나, 최하위 우선순위를 가진 마스터가 오랫동안 버스를 사용하지 못하는 기근(starvation) 현상이 발생하는 것을 방지하기 위한 것임.

여기에는 다음과 같은 알고리즘들이 사용되고 있음.

- * 회전 우선순위(Rotating priority)
- * 임의 우선순위(Random priority)
- * 동등 우선순위(Equal priority)
- * 최소-최근 사용(Least-recently used)

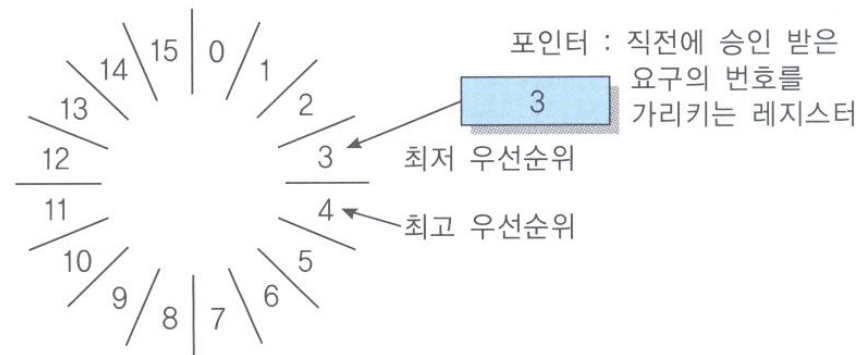
버스 중재 (10)

24

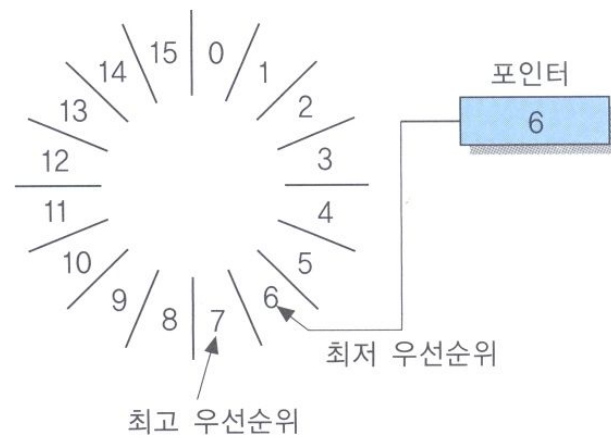
- 회전 우선순위 방식에는, (1) 중재 동작이 끝날 때마다 모든 마스터들의 우선순위가 한 단계씩 낮아지고 가장 우선순위가 낮았던 마스터가 최상위 우선순위를 가지도록 하는 방법과, (2) 일단 버스 사용 승인을 받은 마스터는 최하위 우선순위를 가지며 바로 다음에 위치한 마스터가 최상위 우선순위를 가지도록 하는 방법이 있는데 아래 그림은 두 번째 방식을 나타내고 있음.
- 임의 우선순위 방식에서는 각 중재 동작이 끝날 때마다 마스터들의 우선순위가 임의로 정해지며 이때 각 버스 마스터의 우선순위를 나타내는 숫자는 난수 발생기(random number generator)에 의해 생성됨.
- 동등 우선순위 방식은 비동기식 버스 시스템에서만 사용될 수 있는 방식으로, 모든 마스터들이 동등한 우선순위를 가짐. 이 방식에서 만약 두 개 또는 그 이상의 마스터들이 동시에 버스 요구를 보내면 중재기는 정해진 기준에 의해 그 충돌을 해결 하는데, 일반적으로 FIFO(First-In First-Out) 알고리즘이 사용됨.
- 최소-최근 사용(Least-Recently Used: LRU) 방식은 최근 가장 오랫동안 버스를 사용하지 않은 버스 마스터에게 최상위 우선순위를 할당하는 방식임.

버스 중재 (11)

25



(a) 마스터 3의 요구가 승인된 후



(b) 마스터 6의 요구가 승인된 후

3 교시

버스 중재 (12)

27

(2) 직렬 중재 방식

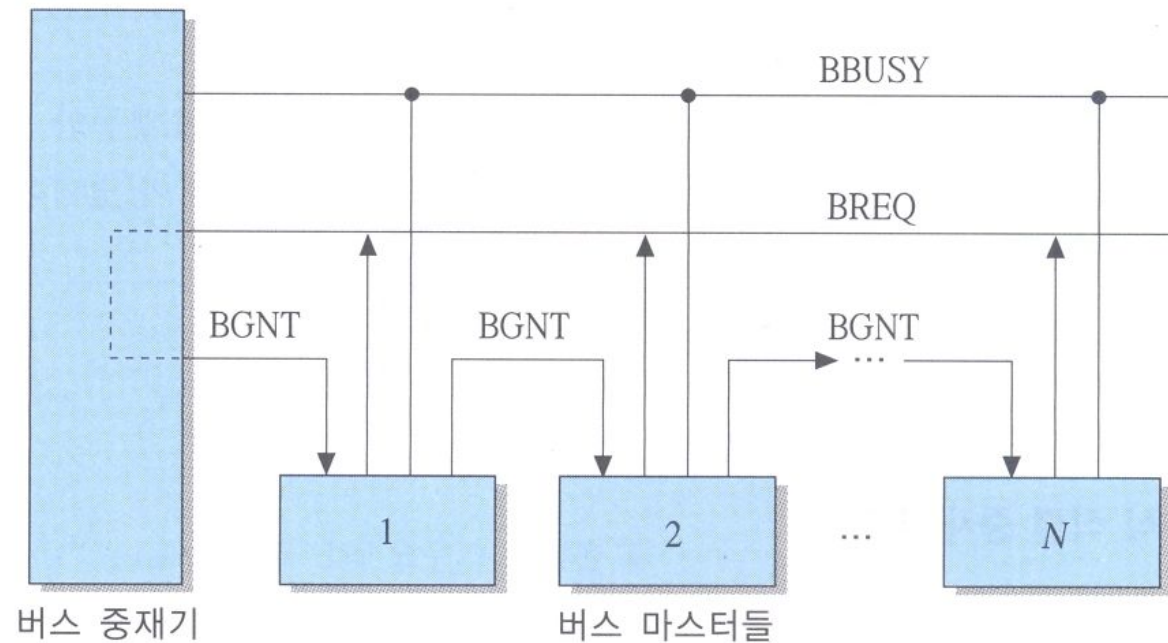
(가) 중앙집중식 직렬 중재 방식

- 직렬 중재 방식의 주요 특징은 하나의 중재 신호선이 모든 버스 마스터들에 직렬로 연결되어 데이지 체인(daisy chain) 형태를 이루는 것임.
그리고 마스터들의 우선순위는 버스 중재기를 시작점으로 하여 승인 신호 선이 연결된 순서대로 정해짐.
- 아래 그림에서 버스 마스터들로부터 발생하는 모든 버스 요구들은 하나의 공통 신호 선을 통해 버스 중재기로 들어옴.
또한 BBUSY 신호도 하나의 공통 신호 선을 사용하며,
어떤 마스터가 버스를 사용하고 있을 때는 세트 되고 그렇지 않은 경우에는 해제 됨.
- 한 개 또는 그 이상의 버스 마스터가 버스 사용을 요구하면,
즉 BREQ 신호가 활성화 되면,
버스 중재기는 데이지 체인의 첫 번째에 접속된 마스터로 승인 신호(BGNT)를 보냄.

버스 중재 (13)

28

버스 중재기에 가장 가까이 위치한 첫 번째 마스터는 가장 먼저 버스 승인 신호를 받기 때문에 버스 사용에 있어서 최상위 우선순위를 가지게 됨. 만약 그 마스터가 버스 요구를 하지 않은 상태에서 승인 신호를 받았다면, 승인 신호를 그 다음에 연결된 마스터로 넘겨줌.



- BGNT 신호가 데이지 체인으로 연결된 중앙집중식 직렬 중재 방식

버스 중재 (14)

29

(나) 분산식 직렬 중재 방식

- 각 버스 마스터가 자신의 중재기를 가진 분산식 직렬 중재 방식은 데이지-체인 버스 승인 신호(daisy-chained bus grant signal: DBGNT)가 버스 중재기들을 순환형으로 접속한 구조로 구성됨.

이 방식에서는 버스 사용권을 부여받은 마스터가 버스 사용을 시작하는 순간에 그 마스터의 중재기가 DBGNT 신호를 세트 하여 자신의 바로 우측에 연결된 마스터의 중재기로 보내줌.

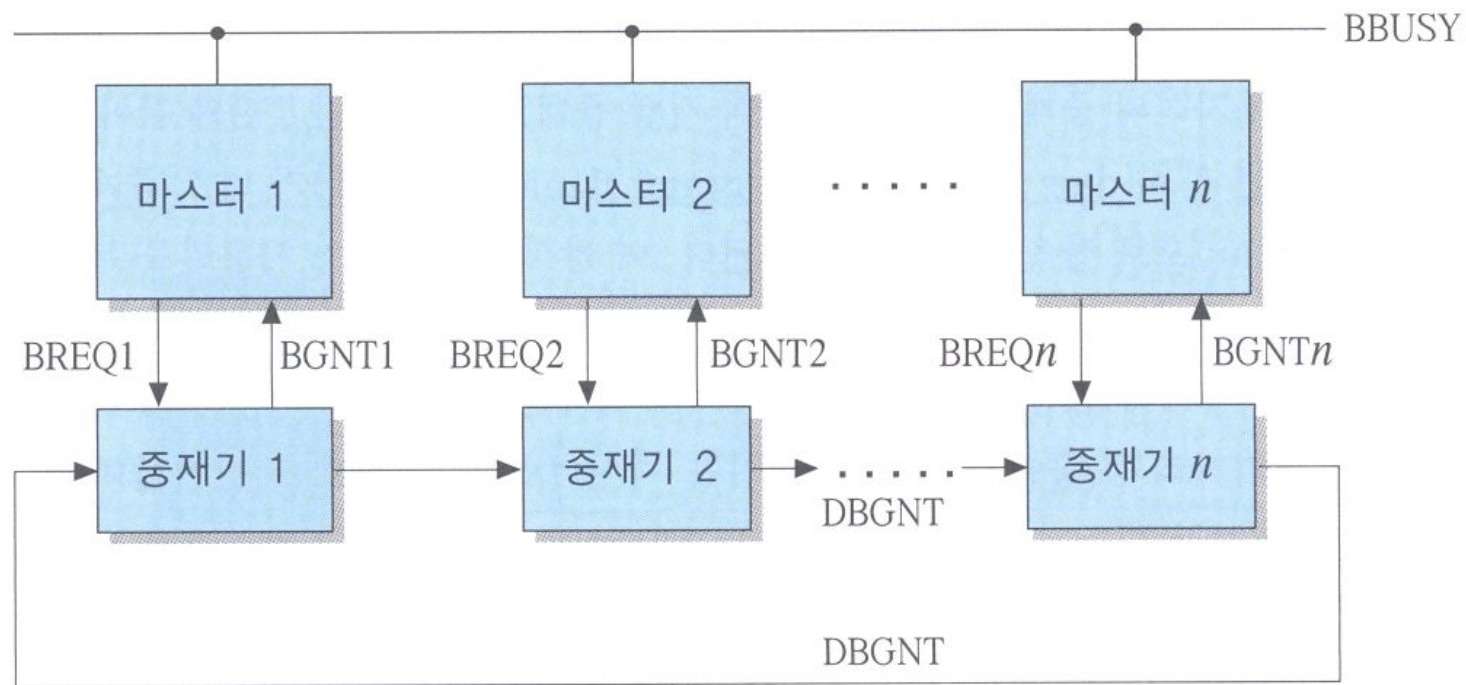
만약 그 마스터가 버스 사용을 신청하고 기다리던 중이었다면, 중재기는 즉시 DBGNT 신호를 받아들여서 BGNT 신호를 발생시켜 마스터로 보냄.

그러나 DBGNT 신호를 받은 마스터가 버스 요구를 하지 않은 상태라면 그 신호를 우측의 다음 중재기로 통과시키고, 그러한 과정은 버스를 요구한 마스터에 도달할 때까지 계속됨.

버스 중재 (15)

30

- 이 방식의 특징은 각 마스터의 우선순위가 계속 변한다는 것임. 즉, 어떤 마스터가 버스 사용 승인을 받으면 그 마스터는 다음 중재 동작에서는 최하위 우선순위를 가지게 되고, 그 마스터의 바로 우측에 위치한 마스터가 최상위 우선순위를 가지게 됨.



- 분산식 직렬 중재 방식의 구성도

버스 중재 (16)

31

(3) 폴링 방식

- 폴링 방식(polling scheme)은 주기적 검사 방식이라고도 하는데, 이 방식에서는 버스 중재기가 각 마스터들이 버스 사용을 원하는지를 주기적으로 검사하여 버스 승인 여부를 결정함.

(가) 하드웨어 폴링 방식

- 이 방식에서는 버스 중재기와 각 버스 마스터 사이에 폴링 동작을 위한 별도의 폴링 선(polling line)이 존재하므로 N개의 마스터를 가진 시스템에는 N개의 폴링 선들이 필요함.

각 선은 중재기가 버스 마스터의 버스 요구 여부를 검사하고자 할 때만 사용하기 때문에 한 번에 한 개씩만 세트 됨.

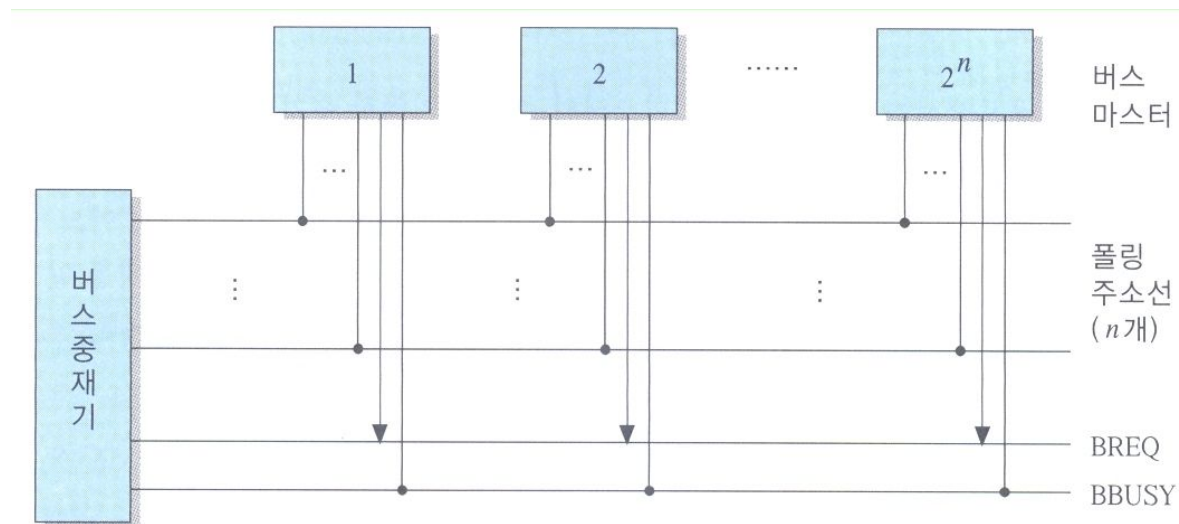
그러나 2진 코드화된 폴링 주소를 사용하면 폴링 선을 \log_2^N 개로 줄일 수 있음.

폴링 선 외에도 공통의 BREQ 선과 BBUSY 선이 각각 한 개씩 필요함.

버스 중재 (17)

32

- 아래 그림은 2진 코드화 된 폴링 주소가 사용되었으며, 동작 순서는 아래와 같음.
 - ① 중재기는 폴링 주소를 발생하여 검사할 마스터를 지정한 다음에, 그 마스터가 버스 사용을 원하는지 물음.
 - ② 지정된 마스터는 버스 사용을 원하면 BREQ 신호를 세트 함.
 - ③ BREQ 신호가 세트 되면, 중재기는 현재 검사 중인 마스터에게 버스 사용을 허가함. 그렇지 않으면(지정된 마스터가 버스 사용을 원하지 않으면), 다음 마스터들에 대한 검사를 순서대로 진행함.



• 하드웨어 폴링 방식

버스 중재 (18)

33

(나) 소프트웨어 폴링 방식

- 이 방식은 하드웨어 폴링 방식과 동일하게 구성되며,
다만 버스 중재기에 프로그램을 실행할 수 있는 간단한 프로세서가 포함되어 있어서
‘지능’을 가지고 있다는 점이 다름.
즉, 중재 동작이 중재기의 고정된 하드웨어에 의해 일률적으로 이루어지는 것이 아니라
프로세서에 의해 조정되는 방식임.
따라서 소프트웨어 폴링 방식은 하드웨어 방식에 비해 속도가 더 느리지만,
융통성이 높다는 장점이 있음.

셀프 테스트

34

- 비동기식 시스템 버스의 읽기 동작에서 기억장치가 데이터 버스에 데이터를 실었다고 CPU에게 응답하는 신호는 무엇인가?

- * SSYN 신호

해설) 비동기식 시스템 버스에서 기억장치가 슬레이브에 해당하므로 데이터버스에 데이터를 실었다는 것을 CPU에게 알리기 위해 SSYN 신호를 사용함.

- 버스 중재의 가변 우선순위 방식에서 우선순위를 결정하기 위해 난수 발생기를 사용하는 방식은 무엇인가?

- * 임의 우선순위 방식

해설) 임의 우선순위 방식은 랜덤 하게 우선순위를 결정하므로 이를 위해 난수 발생기를 사용함.

- 버스 중재기가 각 마스터들이 버스 사용을 원하는지를 주기적으로 검사하여 버스 승인 여부를 결정하는 방식은 무엇인가?

- * 폴링 방식

해설) 폴링 방식에서는 버스 중재기가 주기적으로 각 마스터가 버스 사용을 원하는 지를 조사함.

요점 정리

35

- 시스템 버스의 선들은 전송되는 정보의 종류에 따라 세 가지 버스, 즉 데이터 버스(data bus), 주소 버스(address bus), 제어 버스(control bus) 등으로 나누어짐.
- 시스템 버스에서 모든 버스 동작들이 발생하는 시간이 공통의 버스 클럭을 기준으로 결정되는 버스를 동기식 버스(synchronous bus)라 하고, 버스 동작들의 발생 시간이 다른 버스 동작의 발생 여부에 따라 결정되는 비동기식 버스(asynchronous bus)가 있음.
- 버스 경합이 발생한 경우에 어떤 기준에 따라 버스 마스터들 중에서 한 개씩만 선택하여 순서대로 버스를 사용할 수 있게 해주는 동작을 버스 중재(bus arbitration)라 하며, 버스 중재 방식에는 병렬 중재 방식과 직렬 중재 방식이 있음.
- 버스 중재 방식은 버스 중재기의 위치에 따라 중앙집중식 중재 방식과 분산식 중재 방식이 있음.
- 폴링 방식(polling scheme)은 주기적 검사 방식이라고도 하는데, 이 방식에서는 버스 중재기가 각 마스터들이 버스 사용을 원하는지를 주기적으로 검사하여 버스 승인 여부를 결정함.
- 폴링 방식에는 하드웨어 폴링 방식과 소프트웨어 폴링 방식이 있음.