

【 기술 노트 5 】

마이크로프로세서에서 버스란 무엇인가?

마이크로프로세서를 이용한 시스템에서 CPU와 외부 메모리나 입출력장치 사이에는 버스로 접속되는 구조를 갖는다. 그러면, 이 버스란 무엇이며, 왜 이를 사용하는가? 이것을 아는 것은 마이크로프로세서를 이해하는데 가장 기초가 될 것이다.

버스(bus)란 디지털 회로에서 동일한 기능을 수행하는 많은 신호선들의 집단이라고 정의할 수 있다. 하나하나의 신호들이 개인 교통수단인 택시에 해당하듯이 버스는 대량 교통수단인 버스나 기차에 해당하는 것이다. CPU가 외부에 있는 메모리나 I/O들의 번지를 지정하는데 사용하는 신호선들을 어드레스 버스라고 하며, CPU가 외부에 있는 메모리나 I/O들과 데이터를 주고받는데 사용하는 신호선들을 데이터 버스라고 한다.

컴퓨터 시스템에서 버스 구조를 사용하면 전체적인 회로가 매우 간단해진다. 예를 들어 서울에서 수원이나 대전, 대구, 부산 등으로 가는 고속도로를 각각 별도로 건설한다면 서울 근방의 땅은 말할 것도 없고 전국이 온통 고속도로로 뒤덮일 것이다. 그러나, 서울에서 부산까지 1개의 경부고속도로를 건설하고 이 주변에 있는 도시로 가고자 하는 경우에는 이 고속도로와 가장 가까운 곳에서 인터체인지를 통하여 접속하면 편리할 것이다. 이렇게 함으로써 훨씬 적은 고속도로로도 전국의 각 도시들을 편리하게 연결할 수 있게 될 것이다. 컴퓨터에서 버스란 이와 같이 고속도로의 역할을 한다.

컴퓨터에서는 표준적으로 다음과 같은 3가지의 시스템 버스(system bus)를 사용한다.

1. 어드레스 버스(address bus)

이것은 마이크로프로세서가 외부의 메모리나 입출력장치의 번지를 지정할 때 사용하는 단방향 버스(unidirectional bus)이다. 따라서, 어드레스 버스 신호선의 수는 최대 사용가능한 메모리의 용량이나 입출력장치의 수를 결정한다. Intel 계열의 대부분 마이크로프로세서에서는 같은 번지값으로 메모리 번지(memory address)와 입출력 장치의 번지(I/O port)를 구별하여 지정할 수 있는 isolated I/O 방식을 채택하고 있으며, Motorola 계열의 마이크로프로세서에서는 주로 이를 구별하지 않는 memory-mapped I/O 방식을 채택하고 있다.

2. 데이터 버스(data bus)

마이크로프로세서에서 메모리나 출력장치로 데이터를 출력하거나 반대로 메모리나 입력장치로부터 데이터를 입력할 때 이들 데이터의 전송로로 사용되는 양방향 버스(bidirectional bus)이다. 일반적으로 데이터 버스 신호선의 수는 그 마이크로프로세서의 워드(word) 길이와 같으며, 이것은 마이크로프로세서의 성능(performance)을 결정하는 중요한 요소가 된다.

3. 제어 버스(control bus)

마이크로프로세서가 현재 수행중인 작업의 종류나 상태를 메모리나 입출력 기기에 알려 주는 출력신호와, 외부에서 마이크로프로세서에게 어떤 동작을 취하도록 요구하기 위한 입력신호 등으로 구성되는 단방향 버스이다. 마이크로프로세서의 기능(function)은 이들 제어신호에 의하여 크게 좌우된다.

컴퓨터 시스템에서의 버스에는 이밖에도 로컬버스(local bus)가 있다. 시스템 버스는 CPU의 주변에 임의의 메모리나 I/O장치를 접속하는데 사용하지만, 로컬버스는 전체 시스템의 특정한 부분에서 국지적으로 사용되는 버스를 지칭한다. 예를 들어 퍼스널컴퓨터에서 프린터 인터페이스 회로를 거쳐서 실제로 프린터를 접속하기 위한 25핀의 프린터 콘넥터에 있는 신호들은 로컬버스라 할 수 있고, 하드디스크를 접속하기 위한 플랫 케이블도 로컬버스라고 할 수 있다.

또한, 8051이나 80196과 같이 마이크로컨트롤러 소자의 핀수를 절약하기 위하여 어드레스와 데이터 버스를 시분할다중화시켜 사용하는 방식에서는 외부에서 74LS573과 같은 8비트 래치 소자를 사용하여 이를 분리하는데, 이를 분리한 어드레스 버스와 데이터 버스는 시스템 버스에 해당하지만 이것들을 분리하기 전의 다중화된 AD7~AD0은 로컬버스라고 할 수 있다.

버스의 기본적인 구조는 물론 각 신호들의 상세한 기능이나 동작 타이밍을 이해하는 것은 해당 마이크로프로세서의 하드웨어적인 기능을 익히는데 가장 기본적인 사항이며, 여기에 메모리나 I/O를 확장하는 하드웨어 설계 능력을 갖추는데 필수적인 지식을 제공하기도 한다.

<추가> 버스를 공부할 때는 필연적으로 트리 스테이트 버퍼(3-state buffer)를 함께 익혀 두어야 한다. 즉, 앞에서 예를 든 고속도로와 같은 버스구조의 회로에서는 여러 도시들이 동시에 회로적으로 연결된다면 신호의 충돌이 발생하여 전기적으로 단락현상이 일어날 수 있으므로 이를 절대적으로 피해야 한다. 다시 말하면 서울과 각 도시를 연결하는 자동차들은 항상 서울과 각 도시가 1대1로만 연결되어야 한다는 것이다.

서울에서 출발한 자동차가 각 도시로 나가기 위하여 톨게이트를 통과할 때는 CPU에서 해당 톨게이트의 어드레스와 함께 라이트(write) 신호가 발생되고, 반대로 각 도시에서 서울로 향하는 자동차를 고속도로에 진입시키려면 해당 톨게이트의 어드레스와 함께 리드(read) 신호가 발생되어야 한다.

이러한 어드레스 신호와 리드/라이트 신호는 CPU에서 발생되지만, 이를 이용하여 해당 톨게이트를 열고 닫으려면 밖에서 어드레스 디코더(address decoder) 및 트리 스테이트 버

퍼가 필요하다. 이와 같이 컴퓨터 시스템에서 트리 스테이트 버퍼는 여러개의 장치가 동시에 데이터 버스를 공유하지 못하도록 시시각각으로 조정하는 톨게이트의 역할을 수행하는 것이다. 메모리 소자나 I/O 인터페이스 소자에 있는 칩선택 단자들은 모두 이와 같이 톨게이트에 해당하는 트리 스테이트 버퍼를 제어하는 신호들이다.