

9주 3강

입출력의 제어기법



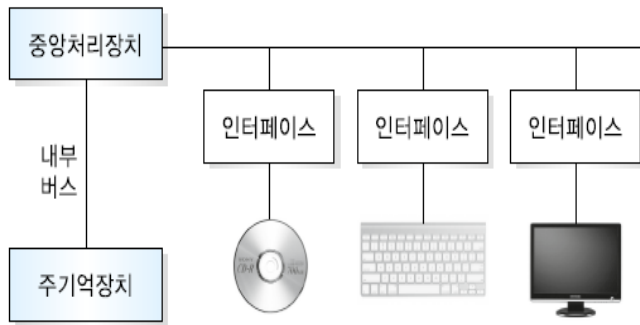


- 입출력장치가 컴퓨터의 내부 장치와 원활한 통신을 수행하려면 통신을 제어할 수 있는 제어 기법이 필요
- 세 가지 형태가 존재
 - 중앙처리장치가 직접 입출력장치를 제어하는 방식
 - 주기억장치와 입출력장치가 직접적으로 데이터를 교환하는 직접 기억장치 액세스(DMA, Direct Memory Access) 방식
 - 별도의 입출력 프로세서가 입출력장치를 제어하는 방법

중앙처리장치가 직접 제어하는 방법



- 데이터 전송, 데이터 상태 검사 등의 모든 것을 CPU가 직접 명령을 수행한다.
- CPU가 제어하는 입출력장치의 구조
 - CPU내의 레지스터에 저장된 내용이 출력장치에 전송되거나 반대로 입력장치에서 내부 레지스터에 전송되고 최종적으로 주기억장치에 저장되도록 하는 방법



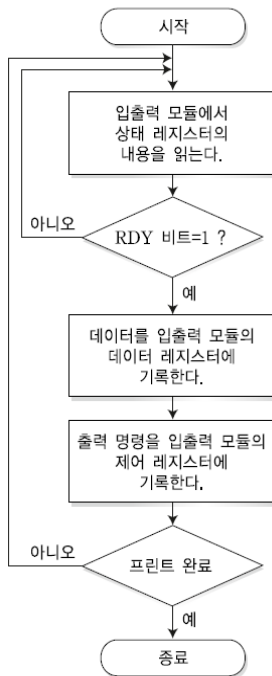
- 중앙처리장치가 제어하는 방법은 다음과 같이 두 가지 방식이 존재
 - 프로그램 입출력(Programmed I/O)
 - 인터럽트- 구동 입출력 (Interrupt-driven I/O)

프로그램 입출력 방식



- 중앙처리장치가 프로그램을 수행하는 도중에 입출력과 관련된 명령을 만나면 해당 입출력 모듈에 명령을 보냄으로써 그 명령을 실행하는 방식

- 프로그램에 의해서 데이터가 출력되는 예의 순서도
 - CPU가 입출력 모듈의 상태 레지스터를 검사해서 출력장치의 상태(RDY 비트)를 판단
 - 과정이 완료될 때까지 상태 검사를 반복하면서 대기, CPU는 다른 작업을 수행할 수 없기 때문에 시간이 낭비.
 - 사용 가능한 상태이면 read/write 명령 중에서 write 명령어를 전송하고 입출력 모듈의 데이터 레지스터에 데이터를 저장한다.
 - CPU는 입출력 모듈에 출력 명령을 레지스터에 기록하여 출력동작을 수행.





- 프로그램 입출력은 입출력 모듈이 데이터를 수신 또는 송신할 준비가 될 때까지 중앙처리장치가 기다려야 한다는 단점이 있다.
- 인터럽트-구동 입출력
 - 입출력 명령을 받은 입출력 모듈이 동작을 수행하는 동안, CPU는 다른 프로그램을 처리 가능
 - CPU가 프로그램 명령어를 실행하는 중이라도 입출력 명령이 있으면 인터럽트를 발생시켜 입출력 동작의 개시를 지시
 - 다시 중앙처리장치는 계속해서 원래의 프로그램 명령을 수행
 - CPU는 입출력이 진행되는 동안 다른 유용한 일을 할 수 있게 된다.



인터럽트(Interrupt)

- 일시 중단의 의미
- CPU가 프로그램을 실행하고 있는 도중에 다른 프로그램을 처리하기 위해 실행 중인 프로그램을 중단 상태로 만들고 다른 프로그램을 처리하는 것.

인터럽트-구동 입출력 방식의 읽기 동작



★ 단계별 동작

- 1단계 : 중앙처리장치가 입출력 모듈로 읽기(read) 명령을 보낸다.
- 2단계 : 입출력 모듈은 주변장치에서 데이터를 읽는다.
이 과정 동안 중앙처리장치는 다른 일을 수행한다.
- 3단계 : 입출력 모듈이 중앙처리장치로 인터럽트 신호를 보낸다.
- 4단계 : 중앙처리장치가 입력된 데이터를 요구한다.
- 5단계 : 입출력 모듈이 중앙처리장치로 데이터를 전송한다.

★ 3단계에서 입출력 모듈이 인터럽트를 요구했을 때, 인터럽트 처리 과정

- ① CPU는 인터럽트에 응답하기 전에 현재 실행 중인 명령어의 실행을 완료.
- ② CPU는 인터럽트를 검사하고 인터럽트 요구가 있다면 요구를 발생한 장치에 확인 신호를 보낸다.
- ③ 확인 신호를 받은 장치는 인터럽트를 요구한 요구 신호를 제거한다.
- ④ CPU는 새롭게 시작될 프로그램으로 제어를 넘겨줄 준비를 한다.
- ⑤ 새로운 프로그램의 시작 주소를 프로그램 카운터에 적재한다.
(인터럽트 서비스를 시작한다)

직접 기억장치 액세스를 이용한 입출력 제어 방식

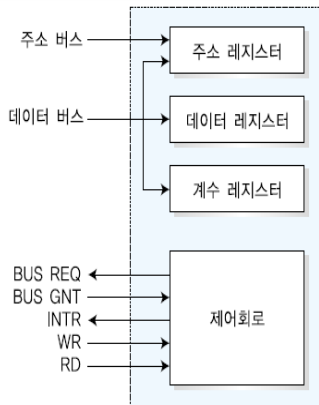


★ 직접 기억장치 액세스(DMA, Direct Memory Access) 방식

- 대용량의 데이터를 이동시킬 때 효과적인 기술
- 기억장치와 입출력 모듈 간의 데이터 전송을 별도의 하드웨어인 DMA 제어기가 처리
- 중앙처리장치는 개입하지 않도록 하는 방식이다.

★ DMA 제어기의 내부 구조

- 주소 레지스터는 주기억장치의 주소를 저장
- 데이터 레지스터는 전송될 데이터를 저장
- 계수 레지스터는 전송되는 데이터 단어의 수를 저장하는 역할
- 제어회로에는 버스 요구 신호(BUS REQ)와 버스 승인신호(BUS GNT), 인터럽트(INTR), 그리고 읽기(WR)와 쓰기(RD)를 위한 연결 단자가 존재한다.





- 1단계 : CPU가 DMA 제어기로 다음 정보를 포함한 명령을 전송
 - 입출력장치의 주소
 - 연산(쓰기 혹은 읽기) 지정자
 - 데이터가 읽혀지거나 쓰여질 주기억장치 영역의 시작 주소
 - 전송될 데이터 단어들의 수
- 2단계 : DMA 제어기는 CPU 로 버스 요구(BUS REQ) 신호를 전송
- 3단계 : CPU 는 DMA 제어기로 버스 승인(BUS GRANT) 신호를 전송
- 4단계 : CPU 의 개입 없이 DMA 제어기가 주기억장치에 데이터를 읽거나 쓴다.
- 5단계 : 전송할 데이터가 남아있으면, 2단계부터 4단계까지 다시 반복
- 6단계 : 모든 데이터의 전송이 완료되면 CPU로 INTR 신호를 전송
- CPU 는 DMA에 명령을 보낸 후에 다른 일을 계속할 수 있게 되며, DMA 제어기가 모든 입출력 동작을 전담하게 된다.
- 중앙처리장치는 전송의 시작과 마지막에만 입출력 동작에 관여

입출력 프로세서를 이용한 입출력 제어방식

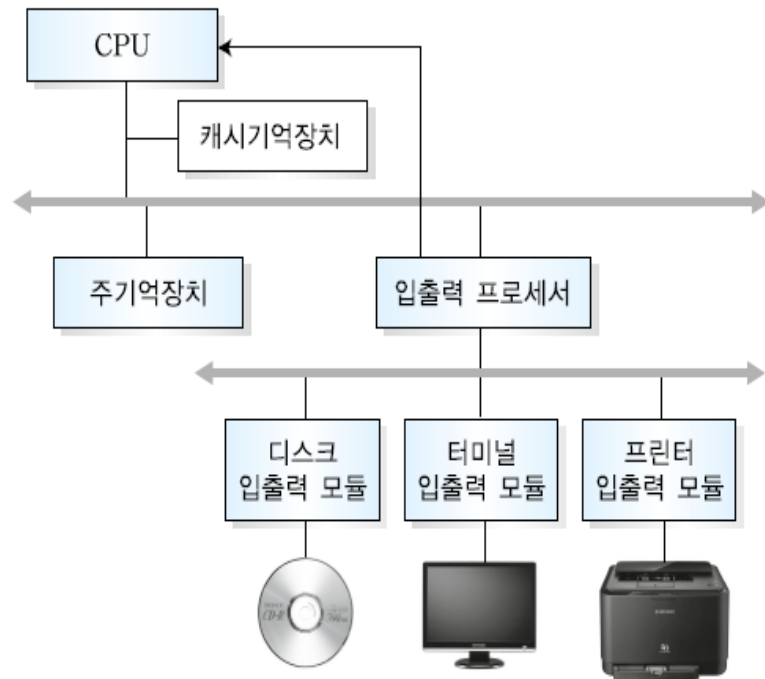


- 입출력 처리를 전담하는 별도의 입출력 프로세서(I/O Processor)를 두어, CPU의 효율을 높이는 입출력 제어방식이다.
- 입출력 프로세서는 DMA 제어기의 기능을 향상시킨 것
 - 입출력 명령어들을 실행할 수 있는 프로세서
 - 데이터 블록을 임시 저장할 수 있는 지역 기억장치를 포함
 - 시스템 버스에 대한 인터페이스 및 버스 마스터 회로와 입출력 버스 중재 회로를 포함
- CPU는 계산 업무에 필요한 데이터만을 처리하고, 입출력 프로세서는 여러 주변장치와 주기억장치 사이의 데이터 전송을 위한 통로를 제공
 - 처음에는 CPU가 입출력 프로세서의 입출력 전송을 시작하도록 하지만, 그 이후에는 CPU와는 독립적으로 입출력 프로세서가 동작한다.
- 입출력 전용 프로세서를 이용해 간섭을 최소화하여 CPU의 이용 효율 증가.
- 별도의 입출력 프로세서로 인한 하드웨어 비용이 증가하는 단점이 있다.

입출력 프로세서를 이용한 시스템



- 입출력 프로세서는 주변장치의 입출력 모듈을 제어해서 독립적으로 데이터를 송신할 수 있다.



다음 시간

10주. 시스템 버스

