

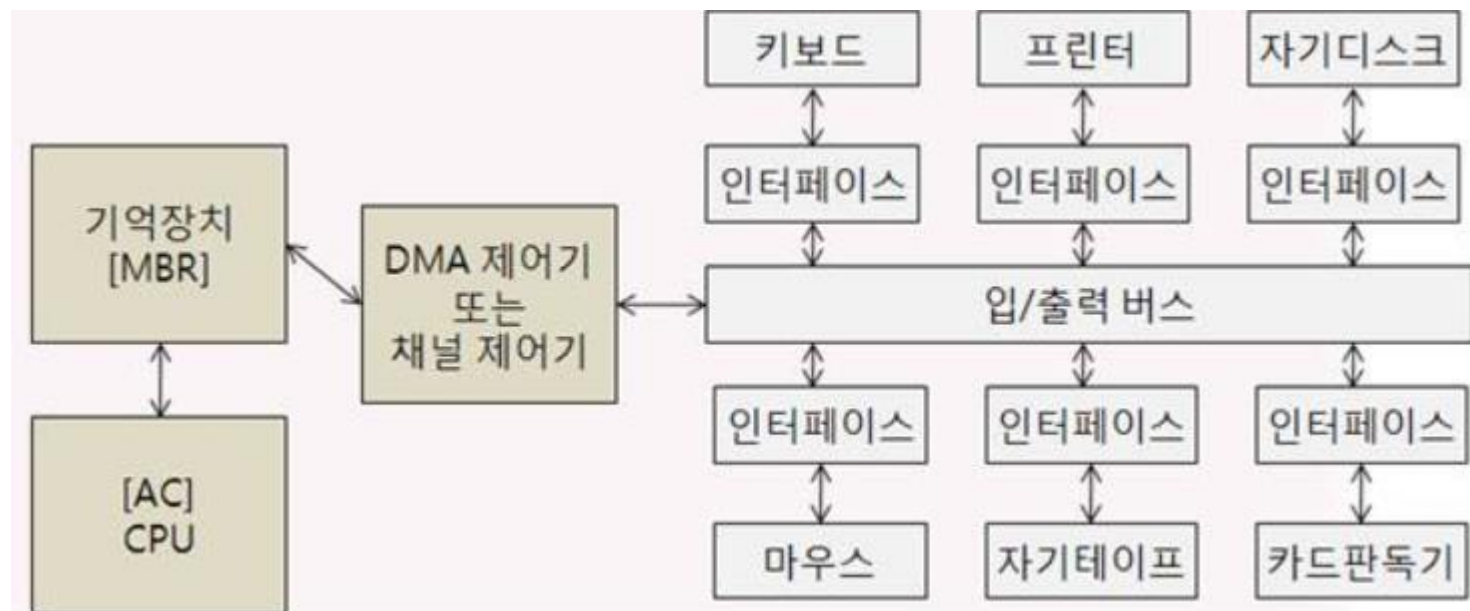
5장 / 입력 및 출력

정보처리산업기사

SECTION 59

입출력의 기본

입출력 장치의 구성



입출력 장치의 구성

- 입출력 제어장치

- 입출력장치와 컴퓨터 사이의 자료 전송을 제어하는 장치이다.
- 데이터 버퍼 레지스터를 이용하여 두 장치 간의 속도 차를 조절한다.
- 제어 신호의 논리적, 물리적 변환 그리고 오류를 제어한다.
- DMA, 채널, 입출력 프로세서, 입출력 컴퓨터 등이 입출력 제어장치에 해당된다.

- 입출력 인터페이스

- 동작 방식이나 데이터 형식이 서로 다른 컴퓨터 내부의 주기억장치나 CPU의 레지스터와 외부 입출력장치 간의 2진 정보를 원활하게 전송하기 위한 방법을 제공한다.
- 컴퓨터와 각 주변장치와의 다음과 같은 차이점을 해결하는 것이 주 목적이다.
 - 전자기 혹은 기계적인 주변 장치와 전자적인 CPU나 메모리 간 동작 방식의 차이
 - 주변 장치와 CPU 간의 데이터 전송 속도의 차이
 - 주변 장치의 데이터 코드와 CPU나 메모리의 워드 형식의 차이
 - 전송 사이클의 길이 등 동작 방식이 서로 다른 주변 장치들의 간섭 없는 제어

입출력 장치의 구성

- 입출력 버스
 - 주기억장치와 입출력장치 사이의 데이터 전송을 위해 모든 주변장치의 인터페이스에 공통으로 연결된 버스이다.
 - 입출력 버스는 데이터 버스, 주소 버스, 제어 버스로 구성된다.
- 입출력 장치의 종류

입력 장치	키보드, 마우스, 스캐너, OMR, OCR, MICR, BCR, 마이크로 필름 입력장치, 라이트 펜, 터치스크린, 디지털타이저 등.
출력 장치	모니터, 프린터, 플로터, 마이크로 필름 출력 장치 등
보조기억장치 (입출력 겸용)	자기 디스크, 자기 테이프, 자기 드럼, 하드 디스크, 플로피디스크 등

기억장치와 입출력장치의 동작 차이

- 기억장치는 처리 속도가 nano(10^{-9})의 단위인 전자적인 장치이고, 입출력장치는 milli (10^{-3})의 단위인 기계적인 장치이므로 동작 방식에는 많은 차이가 있다.

비교 항목	입출력 장치	기억 장치
동작의 속도	느리다	빠르다
동작의 자율성	타율/자율	타율
정보의 단위	Byte(문자)	Word
착오 발생률	많다	적다

비동기 데이터 전송

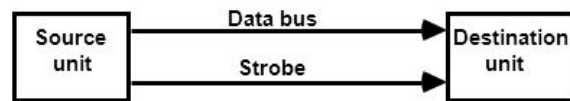
- 두 개의 독립적인 장치 사이의 비동기적인 데이터 전송을 이루기 위해서는 데이터 전송 시각을 알기 위한 제어 신호를 서로 교환하여 송.수신 상태를 서로 맞추어야 한다.
- 비동기 데이터 전송 방식에서 동기를 맞추기 위해 제어 신호를 교환하는 방법에는 스트로브 펄스(Strobe Pulse)에 의한 방식과 핸드셰이킹(Handshaking)에 의한 방식이 있다.

스트로브 펄스(Strobe Pulse) 방식

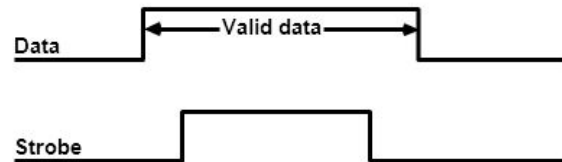
- 데이터 버스와 한 개의 제어선을 이용한다.
- 두 개의 독립적인 장치 사이의 비동기적인 데이터 전송을 이루기 위해 전송시각을 알리는 제어 신호를 스트로브 신호라 하며, 한 개의 제어선을 통해 상호 교환한다.
- 수신 장치는 스트로브 펄스를 발생시켜 송신부로 하여금 데이터를 제공하도록 알린다.
- 메모리와 CPU사이에서 정보를 교환할 때 사용한다.
- 단점 : 전송을 시작한 송신장치는 수신장치가 데이터를 받았는지를 알 수 없다.

Source-Initiated Strobe for Data Transfer

Block Diagram

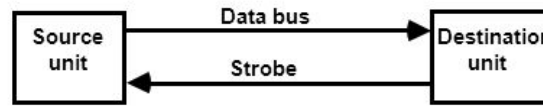


Timing Diagram

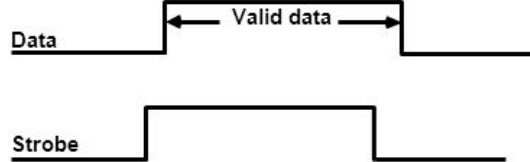


Destination-Initiated Strobe for Data Transfer

Block Diagram

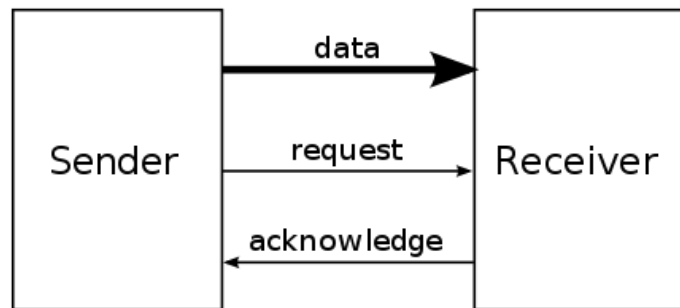


Timing Diagram



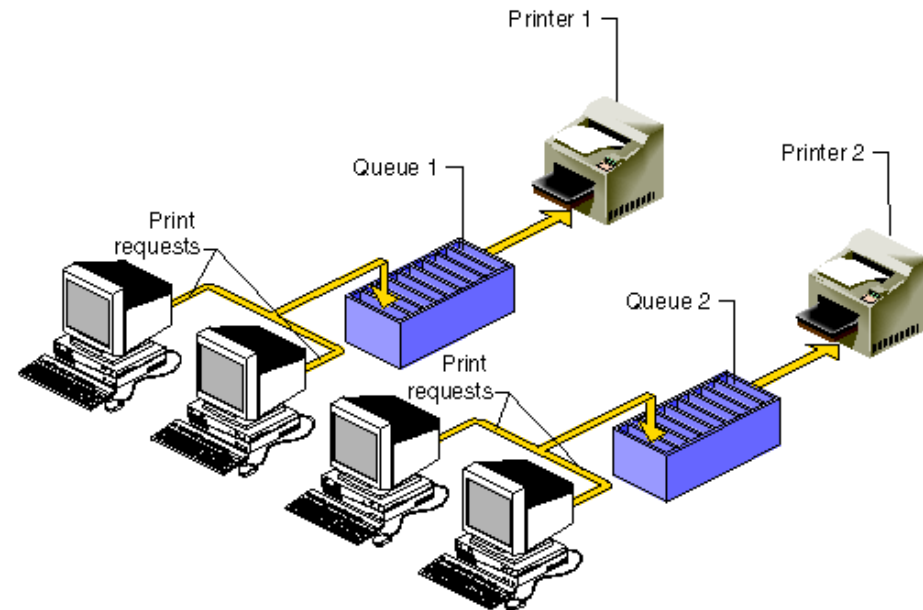
핸드셰이킹(HandShaking) 방식

- 비동기 데이터 전송 방식의 하나로서 데이터 전송 시 송신측과 수신측에서 입출력의 준비나 완료를 나타내는 신호를(RDY, STB)를 사용하여 서로의 동작을 확인하면서 데이터를 전송한다.
- 전송을 시작한 장치에 응답하는 제2의 제어 신호를 전송함으로써 스트로브 제어 방법보다 높은 융통성과 신뢰성을 갖는다.
- 병렬 입출력 데이터 전송 방식의 기본으로 널리 사용된다.
- 2~3개의 제어선을 이용한다.



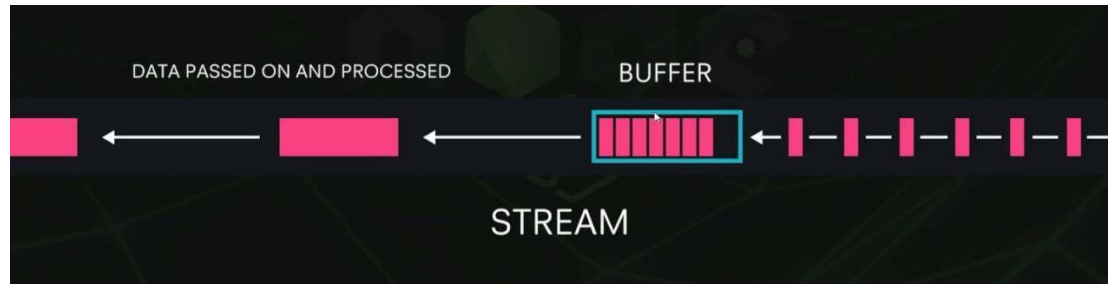
스풀링(Spooling, Simultaneous Peripheral Operation On-Line)

- 스풀링은 다중 프로그래밍 환경하에서 용량이 크고 신속한 액세스가 가능한 디스크를 이용하여 각 사용자 프로그램이 입출력할 데이터를 직접 I/O 장치로 보내지 않고 디스크에 모았다가 나중에 한꺼번에 입출력함으로써 입출력장치의 공유 및 상대적으로 느린 입출력장치의 처리 속도를 보완하는 기법이다.
- 스풀링은 고속의 CPU와 저속의 입출력장치가 동시에 독립적으로 동작하게 하여 높은 효율로 여러 작업을 병행 수행할 수 있도록 해줌으로써 다중 프로그래밍 시스템의 성능 향상을 가져올 수 있다.
- 스풀링은 디스크 일부를 매우 큰 버퍼처럼 사용하며, 큐 방식의 입출력을 수행한다.



버퍼링(buffering)

- 버퍼링도 입출력장치와 CPU 간의 속도 차이를 해결하기 위해 사용하는 목적은 같지만 다음과 같은 점이 스푼링과 다릅니다.



구분	버퍼링	스푼링
저장 위치	주기억장치	보조기억장치
운영 방식	단일 작업	다중 작업
구현 방식	하드웨어	소프트웨어

기출문제

SECTION 60

입출력 제어 방식

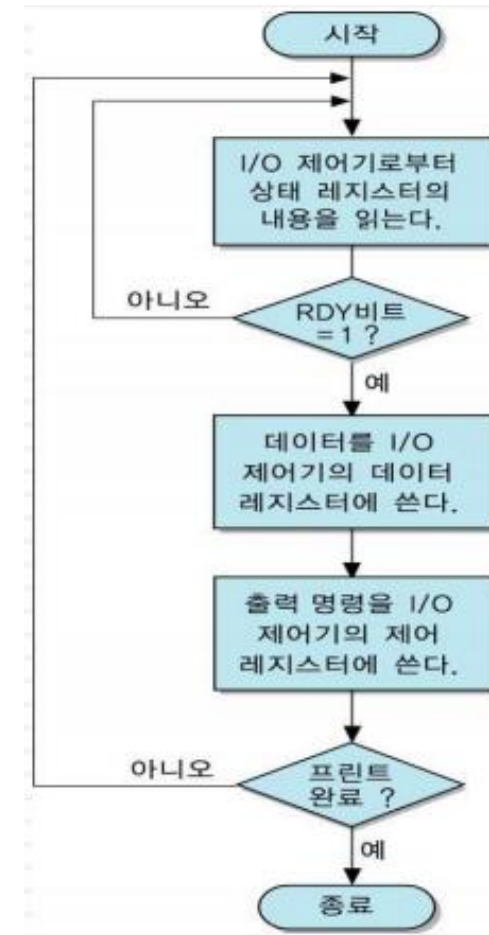
입출력 제어 방식의 개요

- 컴퓨터와 입출력장치 사이의 데이터 전송은 여러 가지 모드로 나누어지는데, CPU를 중간 경로로 이용하여 데이터를 입출력하는 것과 CPU를 거치지 않고 메모리와 직접 입출력하는 것들로 구분된다.

제어 방식	CPU 관여 여부	특징
Program에 의한 I/O	O	가장 원시적인 방식
Interrupt에 의한 I/O	O	
DMA에 의한 I/O	X	소형 컴퓨터에서 이용
Channel에 의한 I/O	X	대형 컴퓨터에서 이용

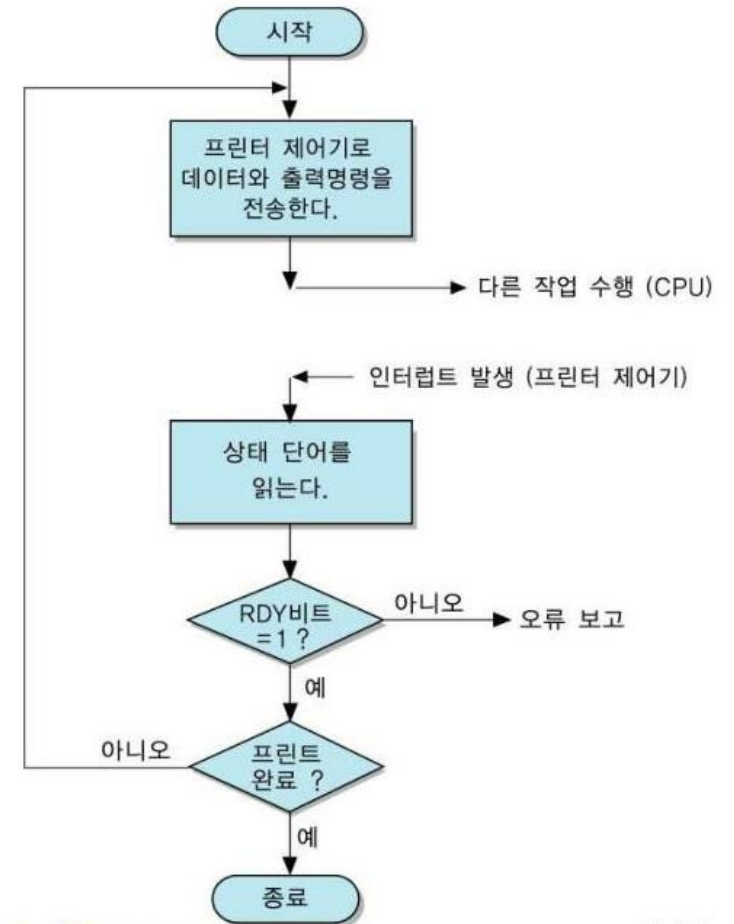
Programmed I/O 방식

- 원하는 I/O가 완료되었는지의 여부를 검사하기 위해서 CPU가 상태 Flag를 계속 조사하여 I/O가 완료되었으면 MDR(MBR)과 AC사이의 자료 전송 속도도 CPU가 직접 처리하는 I/O 방식이다.
- 출력에 필요한 대부분의 일을 CPU가 해주므로 ...
- I/O 작업 시 CPU는 계속 I/O 작업에 관여해야 하기 때문에 다른 작업을 할 수 없다는 단점이 있다.



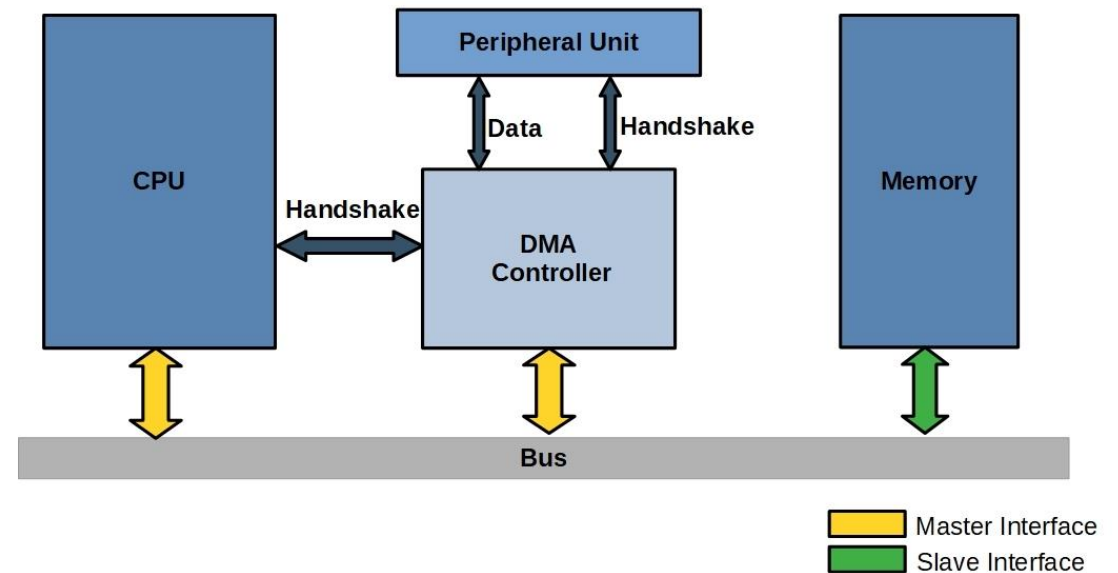
Interrupt I/O

- Interrupt I/O 방식은 입출력을 하기 위해 CPU가 계속 Flag를 검사하지 않고, 데이터를 전송할 준비가 되면 입출력 인터페이스가 컴퓨터에게 알려 입출력이 이루어지는 방식이다.
- 입출력 인터페이스는 CPU에게 인터럽트 신호를 보내 입출력이 있음을 알린다.
- CPU는 작업을 수행하던 중 입출력 인터럽트가 발생하면 수행중인 프로그램을 중단하고 입출력을 처리한 후 원래의 작업으로 돌아와 작업을 계속 수행한다.
- CPU가 계속 Flag를 검사하지 않아도 되기 때문에 Programmed I/O보다 효율적이다.
- 대량의 자료 전송 시 CPU의 부담을 증가시킨다.



DMA(Direct Memory Access)에 의한 I/O (1/2)

- DMA는 입출력 장치가 직접 주 기억장치를 접근하여 Data Block을 입출력 하는 방식으로, 입출력 전송이 CPU의 레지스터를 경유하지 않고 수행된다.
- CPU는 I/O에 필요한 정보를 DMA 제어기에 알려서 I/O 동작을 개시시킨 후 I/O동작에 더 이상 간섭하지 않고 다른 프로그램을 할당하여 수행한다.
- DMA 방식은 입출력 자료 전송 시 CPU를 거치지 않기 때문에 CPU의 부담이 없어 보다 빠른 데이터의 전송이 가능하다.
- DMA는 인터럽트 신호를 발생시켜 CPU에게 입출력 종료를 알린다.
- DMA는 블록으로 대용량의 데이터를 전송할 수 있다.
- DMA는 Cycle Steal 방식을 이용하여 데이터를 전송한다.
- CPU와 DMA 제어기는 메모리와 버스를 공유한다.



DMA(Direct Memory Access)에 의한 I/O (2/3)

- CPU에서 DMA 제어기로 보내는 자료
 - I/O 장치의 주소
 - 데이터가 있는 주 기억장치의 시작 주소
 - DMA를 시작시키는 명령
 - 입출력 하고자 하는 자료의 양
 - 입력 또는 출력을 결정하는 명령
- DMA의 구성 요소
 - 인터페이스 회로 : CPU와 입출력 장치와의 통신 담당
 - 주소 레지스터(Address Register) 및 주소 라인 : 기억장치의 위치 지정에 위한 번지 기억 및 전송
 - 워드 카운트 레지스터(Word Count Register) : 전송되어야 할 워드의 수 기억
 - 제어 레지스터(Control Register) : 전송 방식 결정
 - 데이터 레지스터(Data Register) : 전송에 사용할 자료나 주소를 임시로 기억하는 버퍼 역할을 함

DMA(Direct Memory Access)에 의한 I/O (3/3)

○ DMA의 전송 절차

- CPU가 DMA 제어기에 명령을 내린다.
- DMA 제어기가 CPU에게 버스 사용을 요구한다.
- CPU가 DMA 제어기에 버스 사용을 허가한다.
- DMA 제어기가 주 기억장치에서 데이터를 읽어 디스크로 전송한다.
- 2~4번을 반복하다가 데이터 전송이 완료되면 인터럽트 신호를 보낸다.

○ Cycle Steal

- 데이터 채널(DMA 제어기)과 CPU가 주 기억 장치를 동시에 Access할 때 우선순위를 데이터 채널에게 주는 방식이다.
- Cycle Steal은 한 번에 한 데이터 워드를 전송하고 버스의 제어를 CPU에게 돌려준다.
- Cycle Steal을 이용하면 입출력 자료의 전송을 빠르게 처리해 주는 장점이 있다.
- Cycle Steal시 중앙처리장치는 메모리 참조가 필요 없는 오퍼레이션을 계속 수행한다.

Channel에 의한 I/O

- Channel은 I/O를 위한 특별한 명령어를 I/O 프로세서에게 수행토록 하여 CPU관여 없이 주 기억장치와 입출력장치 사이에서 입출력을 제어하는 입출력 전용 프로세서이다.
- DMA 제어기의 한계를 극복하기 위하여 고안된 방식이다.
- 채널은 DMA 방법으로 입출력을 수행하므로 DMA의 확장된 개념으로 볼 수 있다.
- 채널의 특징
 - 채널 제어기는 채널 명령어로 작성된 채널 프로그램을 해독하고 실행하여 입출력 동작을 처리한다.
 - 채널은 CPU로부터 입출력 전송을 위한 명령어를 받으면 CPU와는 독립적으로 동작하여 입출력을 완료한다.
 - 채널은 주 기억장치에 기억되어 있는 채널 프로그램의 수행과 자료의 전송을 위하여 주기억장치에 직접 접근한다.
 - I/O 장치는 제어장치를 통해 채널과 연결된다.
 - I/O 채널은 CPU의 I/O 명령을 수행하지 않고 I/O 채널 내의 특수목적 명령을 수행한다.
 - 채널은 CPU와 인터럽트로 통신한다.

Channel에 의한 I/O

채널과 DMA의 비교

DMA 방식과의 유사점	-CPU에 대해 완전히 자율적으로 동작, 즉 CPU는 I/O 동작을 개시시킨 후 I/O 동작에 더 이상 간섭하지 않고 다른 명령을 수행한다. -주 기억장치를 직접 접근하여 I/O 명령을 수행하고 I/O 관련 자료를 기록하거나 판독한다.
DMA 방식과의 차이점	-DMA는 한 개의 Instruction에 의해 한 개의 Block만을 입출력한다. -Channel은 한 개의 Instruction에 의해 여러 개의 Block을 입출력한다.

채널의 종류

Selector Channel (선택 채널)	-고속 입출력장치(자기 디스크, 자기 테이프, 자기 드럼)와 입출력하기 위해 사용한다. -특정한 한 개의 장치를 독점하여 입출력한다.
Multiplexer Channel (바이트 다중 채널)	-저속 입출력장치(카드리더, 프린터)를 제어하는 채널이다. -동시에 여러 개의 입출력장치를 제어한다.
Block Multiplexer Channel (블록 다중)	-고속 입출력장치를 제어하는 장치이다. -동시에 여러 개의 입출력장치를 제어한다.

기출문제

SECTION 61

인터럽트의 개념

인터럽트

- 인터럽트는 프로그램을 실행하는 도중에 예기치 않은 상황이 발생할 경우, 현재 실행중인 작업을 즉시 중단하고 발생한 상황을 우선 처리한 후 실행중이던 작업으로 복귀하여 계속 처리하는 것을 말한다. 일명 "끼어들기" 라고도 한다.
- 인터럽터는 외부 인터럽트, 내부 인터럽트, 소프트웨어 인터럽트로 분류하는데, 외부나 내부 인터럽트는 CPU의 하드웨어에서의 신호에 의해 발생하고 소프트웨어 인터럽트는 명령어의 수행에 의해 발생한다.
- 외부 인터럽트
 - 전원 이상 인터럽트(Power Fail Interrupt) : 정전이 되거나 전원 이상이 있는 경우
 - 기계 착오 인터럽트(Machine Check Interrupt) : CPU의 기능적인 오류 동작이 발생한 경우
 - 외부 신호 인터럽트(External Interrupt) : 타이머에 의해 규정된 시간을 알리는 경우, 키보드로 인터럽트 키를 누른 경우, 외부 장치로부터 인터럽트 요청이 있는 경우
 - 입출력 인터럽트(Input-Output Interrupt) : 입출력 Data의 오류나 이상 현상이 발생한 경우, 입출력장치가 데이터의 전송을 요구하거나 전송이 끝났음을 알릴 경우

인터럽트

○ 내부 인터럽트

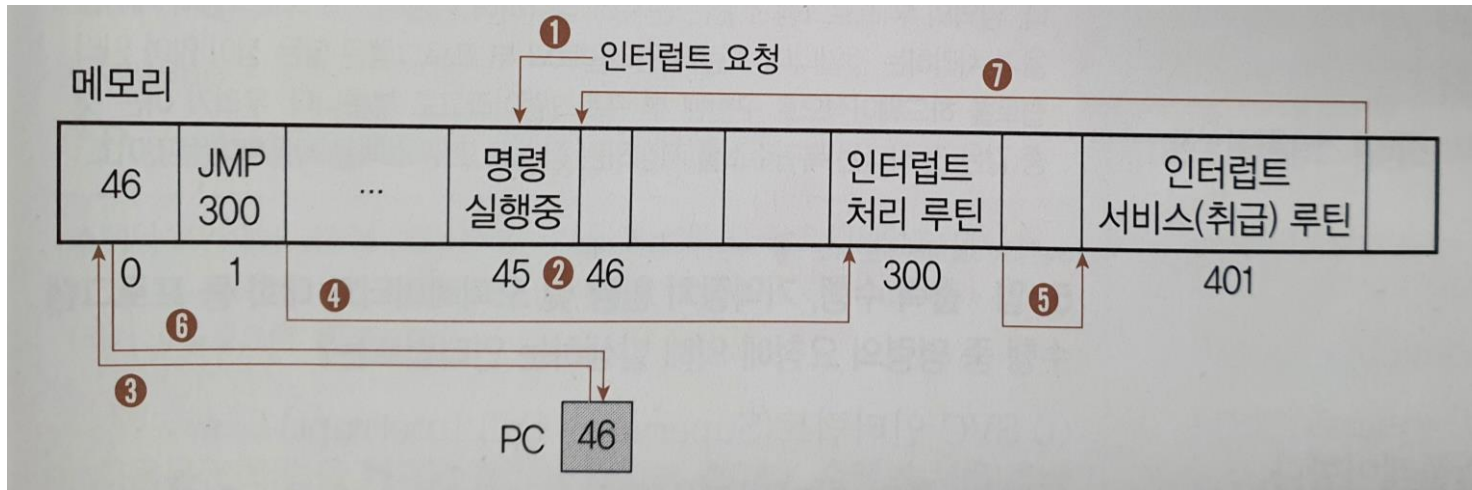
- 다음과 같이 잘못된 명령이나 데이터를 사용할 때 발생하며, 트랩(Trap) 이라고도 부른다.
- 프로그램 검사 인터럽트(Program Check Interrupt)
 - 0으로 나누기 어려운 경우
 - OverFlow 또는 UnderFlow가 발생한 경우
 - 프로그램에서 명령어를 잘못 사용한 경우
 - 부당한 기억장소의 참조와 같은 프로그램 상의 오류

○ 소프트웨어 인터럽트

- 프로그램 처리중 명령의 요청에 의해 발생하는 것으로, 가장 대표적인 형태는 감시 프로그램을 호출하는 SVC(SuperVisor Call) 인터럽트가 있다.
 - 사용자가 SVC 명령을 써서 의도적으로 호출한 경우
 - 복잡한 입출력 처리를 해야 하는 경우
 - 기억장치 할당 및 오퍼레이터와 대화를 해야 하는 경우

인터럽트

- 인터럽트 발생시 CPU가 확인할 사항
 - 프로그램 카운터의 내용
 - 사용한 모든 레지스터의 내용
 - 상태 조건의 내용(PSW)
- 인터럽트의 동작원리



기출문제

SECTION 62

인터럽트 우선순위(Priority) 체제

인터럽트 우선 순위



- 우선순위 인터럽트 체제의 목적은 CPU에 연결되어 있는 여러 장치에서 동시에 하나 이상의 인터럽트가 발생하였을 때 먼저 서비스할 장치를 결정하기 위해서이다.
- 인터럽트 우선순위 체제의 기능
 - 각 장치에 우선 순위를 부과하는 기능
 - 인터럽트를 요청한 장치의 우선순위를 판별하는 기능
 - 우선순위가 높은 것을 먼저 처리할 수 있는 기능
- 인터럽트 우선순위를 판별하는 방법은 소프트웨어적인 방법과 하드웨어적인 방법이 있다.

소프트웨어적인 인터럽트 우선순위 판별 방법

Polling

- 소프트웨어적인 방법은 Interrupt 발생 시 우선순위가 가장 높은 인터럽트 자원부터 인터럽트 요청 플래그를 차례로 검사하여 찾고 이에 해당하는 인터럽트 서비스 루틴을 수행하는 방식이다.
- 소프트웨어적인 방식을 폴링이라고 한다.
- 우선순위 변경이 쉬우며, 자기디스크와 같이 속도가 빠른 장치에 높은 동급을 부여한다.
- 회로가 간단하고 융통성이 있으며 별도의 하드웨어가 필요 없으므로 경제적이다.
- 많은 인터럽트가 있을 때 그들을 모두 조사하는 데 많은 시간이 걸려 반응시간이 느리다는 단점이 있다.

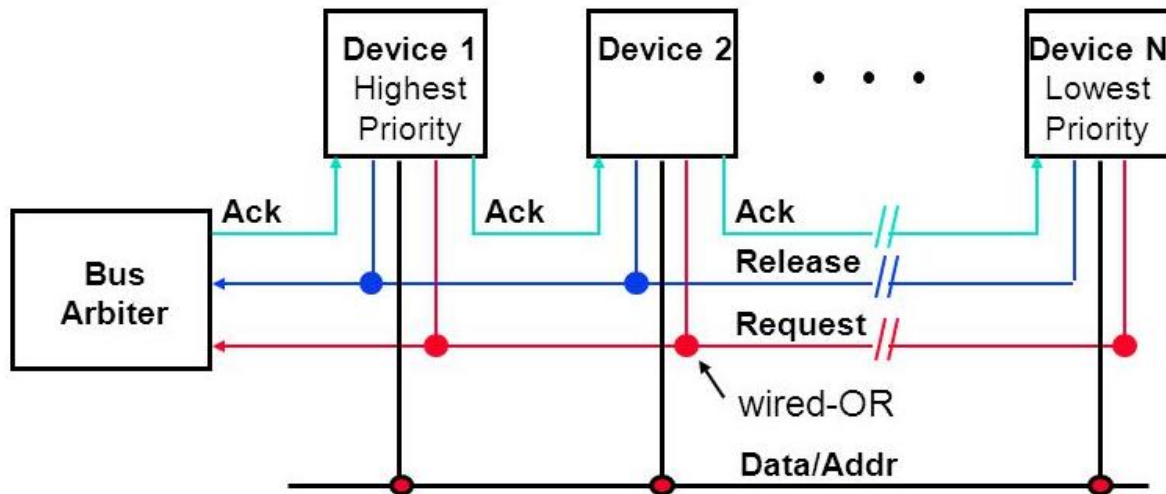
하드웨어적인 인터럽트 우선순위 판별 방법

Vectored Interrupt

- 하드웨어 우선순위 인터럽트는 CPU와 Interrupt를 요청할 수 있는 장치 사이에 장치 번호에 해당하는 버스를 병렬이나 직렬로 연결하여 요청 장치의 번호를 CPU에 알리는 방식이다.
- 하드웨어적인 방법을 벡터 인터럽트(Vectored Interrupt) 라고 한다.
- 벡터 인터럽트 방식에서는 인터럽트를 발생한 장치가 프로세서에게 분기할 곳에 대한 정보를 제공하는데, 이 정보를 인터럽트 벡터라 한다.
- 하드웨어적인 방법은 장치 판별을 위한 별도의 프로그램 루틴이 없어 응답 속도가 빠르다.
- 회로가 복잡하고 융통성이 없으며 추가적인 하드웨어가 필요하므로 비경제적이다.
- 하드웨어적인 방법은 직렬과 병렬 우선순위 부여 방식이 있다.

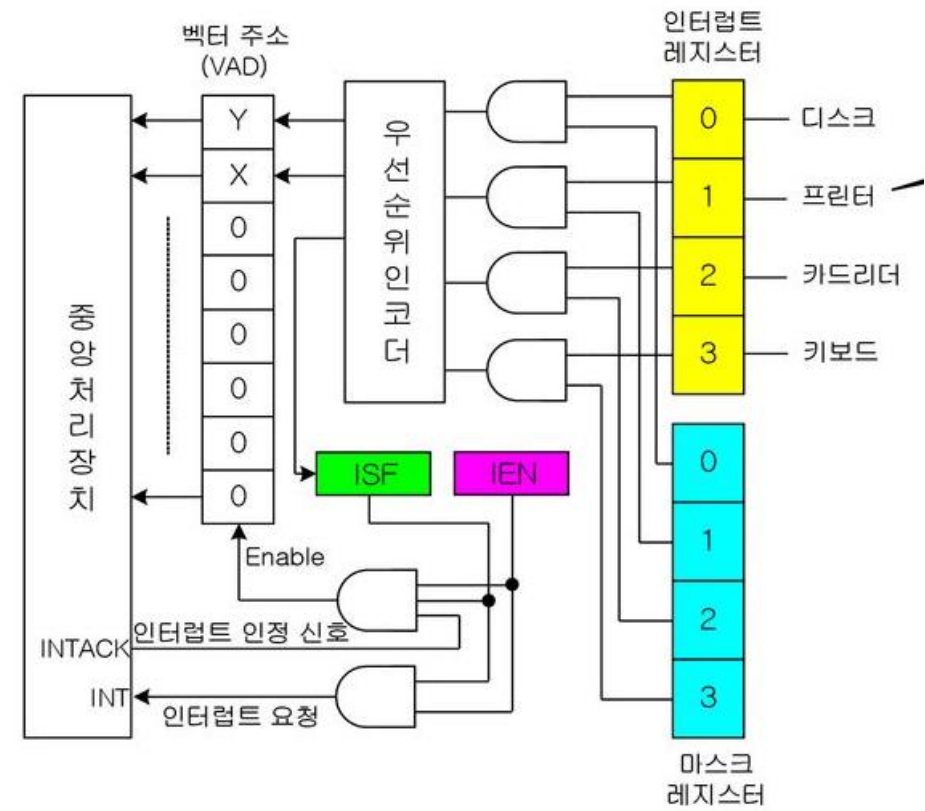
직렬(Serial) 우선순위 부여 방식 : 데이지 체인 (Daisy-Chain) 방식

- 직렬 우선순위 부여 방식은 인터럽트가 발생하는 모든 장치를 한 개의 회선에 직렬로 연결한다.
- 우선순위가 높은 장치를 선두에 위치시키고 나머지를 우선 순위에 따라 차례로 연결한다.
- 직렬 우선 순위 부여 방식을 데이지 체인 방식이라고 한다.



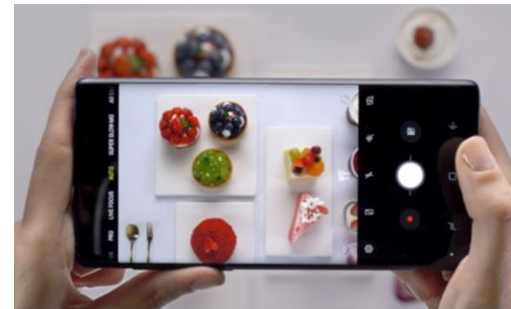
병렬(Parallel) 우선순위 부여 방식

- 병렬 우선순위 부여 방식은 인터럽트가 발생하는 각 장치를 개별적인 회선으로 연결한다.
- 각 장치의 인터럽트 요청을 제어하기 위해 각 비트를 개별적으로 Set할 수 있는 Mask Register를 사용한다.
- 우선순위는 Mask Register의 비트 위치에 의해 결정된다.
- 마스크 레지스터는 우선순위가 높은 것이 서비스 받고 있을 때 우선순위가 낮은 것을 비활성화 시킬 수 있다.
- 우선순위가 높은 Interrupt는 낮은 Interrupt가 처리되는 중에도 우선 처리된다.



하드웨어적인 방법과 소프트웨어적인 방법의 특징 비교

비교 항목	하드웨어	소프트웨어
반응속도	고속	저속
회로 복잡도	복잡	간단
경제성	비경제적	경제적
융통성	없음	있음



기출문제