하드 디스크

1. 외부 저장장치 유형

- **Magnetic disk**
  - HDD

- **Optical memory**
  - CD-ROM
  - CD-Recordable (CD-R)
  - CD-R/W
  - DVD

- **Magnetic tape**

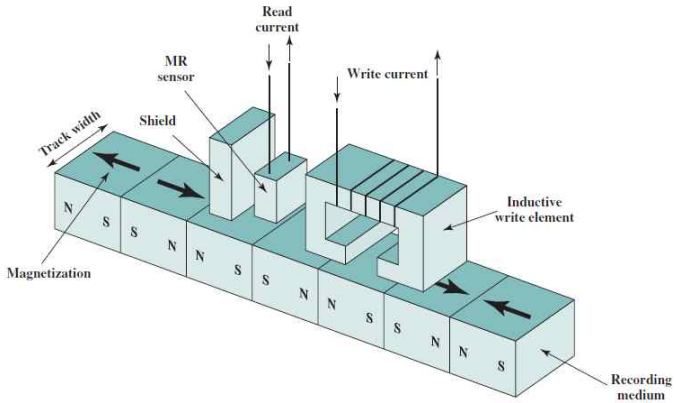
2. HDD

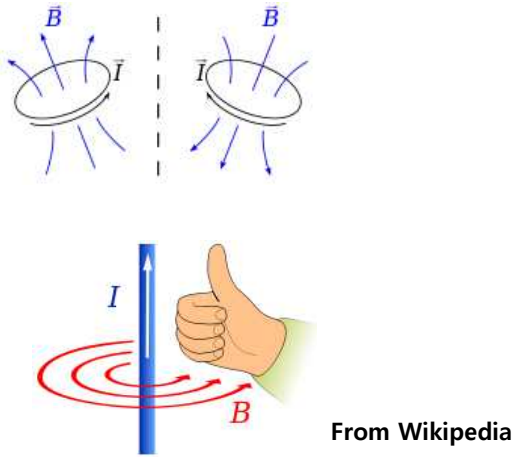
- **HDD(Hard Disk Drive)**
  - 저렴한 비용으로 큰 용량을 확보할 수 있는 외부 기억장치.
  - 자성물질로 코팅된 비자성 원형 평판위에 구성된 기억장치.
    - 평판(Substrate)재료 : 알루미늄, 알루미늄 합금, 유리

- **구성품**
  - 원형 평판(Circular Platter) : 자성물질로 코팅된 평판, 데이터 저장장소.
  - 헤드(Head) : 전도성 코일을 사용하여 데이터를 기록하거나 판독하는 장치
  - 디스크 암(Disk Arm) : 헤드를 이동시키는 장치
  - 구동장치(Actuator) : 디스크 암을 움직이게 하는 모터.

2.1 HDD의 데이터 읽기/쓰기

- **Data Read/Write**
  - 데이터는 헤드 (head) 라고 불리는 전도성 코일 (conductive coil) 을 통하여 디스크에 쓰거나 읽혀진다.





From Wikipedia

2.1 HDD의 데이터 읽기/쓰기

- **데이터 쓰기**
  - 전도성 코일에 전류가 흐르면 자기장을 형성하는 원리를 이용.
  - 헤드에 펄스를 보내 다른 형태의 자성 패턴을 표면에 기록.

- **데이터 읽기**
  - 자기장 내에 코일이 지나가면 유도전류가 발생하는 원리를 이용.
  - MR(Magneto Resistive) 센서를 사용한 Read
    - 움직이는 물체의 자화 방향에 따라 전기 저항이 변하는 원리를 이용
    - MR 센서에 전류를 흘려주면, 자화 방향에 따라 전압이 달라진다.
    - 고속 동작이 가능 → 저장 밀도와 동작속도가 향상.

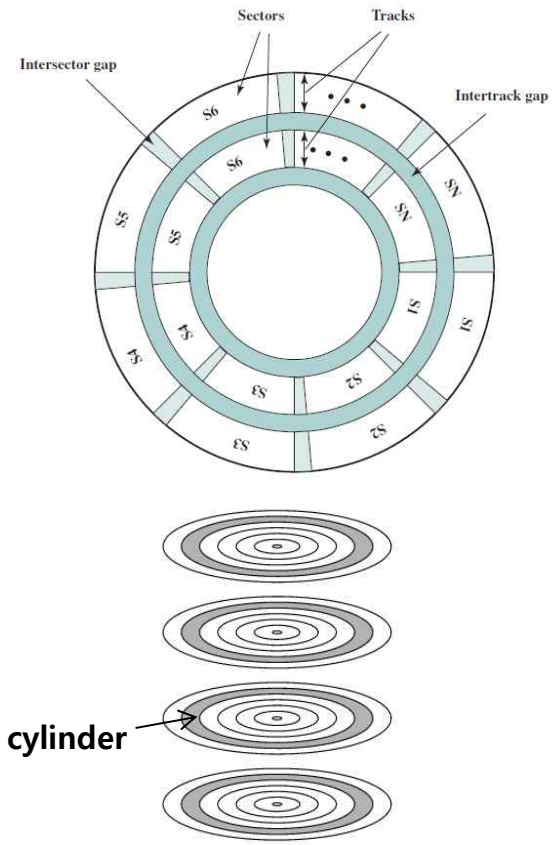
2.2 HDD 논리 구성

- **디스크 표면을 트랙과 섹터로 나누어서 데이터를 저장**

- **트랙(Track)**
  - 디스크 표면에 동심원 형태로 분리한 영역

- **섹터(Sector)**
  - 트랙을 일정 크기(512 Bytes)로 분할한 영역
  - 물리적인 최소 단위

- **실린더(Cylinder)**
  - 원형 평판(Platter)를 여러 개 사용하는 경우, 각 평판들의 동일 트랙으로 구성하는 영역



2.2 HDD 논리 구성

- **클러스터(Cluster)**
  - 운영체제가 디스크에 데이터를 읽고 쓰는 논리적 기본단위
    - 1개 이상의 섹터를 묶어서 소프트웨어적으로 관리.
    - 클러스터 크기 : 4~32KB
  - 사용하는 파일 시스템에 따라 클러스터 크기 다양.
  - 대용량 데이터 저장에 효율적, 소량의 데이터에는 비효율적

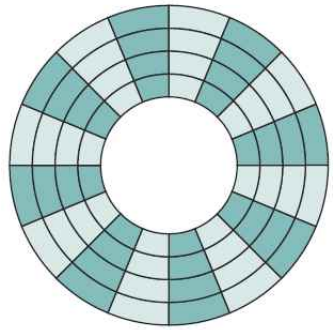
- **블록(Block)**
  - 운영체제에서 사용하는 최데 데이터 전송단위
  - 메모리와 디스크 사이의 입출력 전송단위
  - 여러 개의 sector로 구성
    - 4KB

2.3 디스크 레이아웃 방법

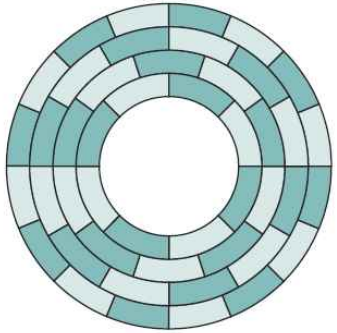
- **회전하는 디스크의 바깥쪽에 위치한 데이터는 헤드를 지나는 속도가 중심부에 위치한 데이터보다 빠르다.**
  - 트랙별로 비트간의 간격을 증가시킨다.

- **등각속도 (CAV : constant angular velocity)**
  - 디스크를 고정속도로 회전시킨다.
    - 같은 비율로 비트를 액세스 가능.
  - 단점 : 바깥쪽 트랙의 공간 낭비
    - 데이터 밀도 저하.

- **데이터 밀도를 증가시키기 위해 MZR (multiple zone recording) 사용.**
  - 디스크의 표면을 여러 개의 영역 (zone) 으로 나눈다.
  - Zone 내에서 트랙당 비트 수는 동일
  - 동일 트랙 길이.
  - 단점 : 복잡한 회로 필요



CAV



MRZ

2.4 디스크 어드레싱

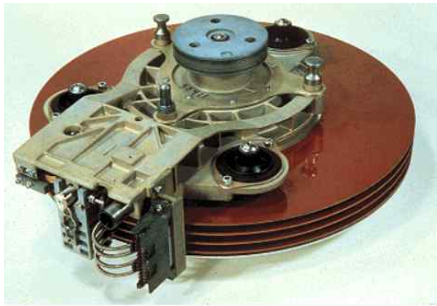
- **하드 디스크의 최소 기록 단위는 섹터**
  - 섹터의 위치를 지정하는 주소지정방식 필요
  - 주소 지정방식 : CHS, LBA

- **CHS(Cylinder-Head-Sector) 주소 지정방식**
  - 하드 디스크의 구조와 동일하게 실린더 번호, 헤드 번호, 섹터 번호를 사용하여 섹터를 지정하는 방식
  - 주소 지정에 많은 비트 필요 → 대용량 하드 디스크에 부적합

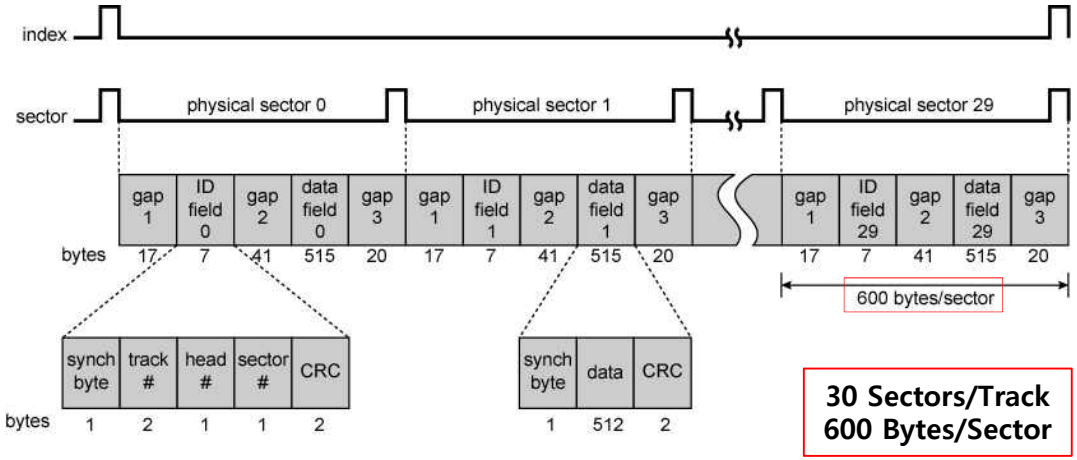
- **LBA(Logical Block Addressing) 주소 지정방식**
  - 하드 디스크의 모든 섹터를 1차원 배열로 생각하고, 0번부터 차례로 연속적인 논리주소를 할당하는 방식.
  - 디스크 컨트롤러에서 논리주소를 물리주소(C-H-S번호)로 변환한다.
  - 운영체제는 하드 디스크를 섹터의 1차원 배열로 취급.

2.5 디스크 포맷

- **Winchester 디스크 포맷**
  - IBM Winchester 디스크에 적용된 디스크 포맷.
    - 30 Sectors/Track
    - 600 Bytes/Sector (512Byte Data)



The Winchester Disk



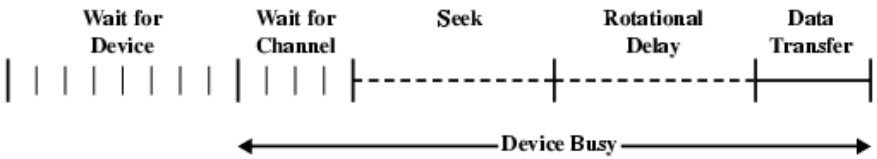
2.6 디스크 포맷팅

- **디스크 포맷팅(Formatting)**
  - 하드 디스크에 데이터 저장하기 위한 준비 작업
    - 트랙과 섹터의 시작과 끝을 식별하기 위하여 디스크에 제어 데이터를 기록

- **디스크 포맷 유형**
  - Low-level 포맷
    - 실린더, 트랙과 섹터 위치를 잡아주는 과정.
    - 시간이 많이 소요.
    - 모든 섹터의 내용을 삭제하고 공장 출고 상태로 초기화.
  - High-level 포맷
    - 운영체제가 사용하는 파일 시스템을 설정하는 과정.
    - 운영체제 설치과정에서 실행
    - 명령어를 사용해서 직접 실행.

2.7 디스크 성능 파라미터

- 탐색 시간(Seek Time) : 헤드가 원하는 트랙으로 이동하는데 걸린 시간
- 회전 지연 시간(Rotational Latency) : 원하는 섹터가 헤드 밑에 도달하기 까지 걸린 시간.
- 접근 시간(Access Time) : Seek Time + Rotational Delay
- 전송 시간(Transfer Time) : 데이터 전송을 위해 걸리는 시간.



2.7 디스크 성능 파라미터

- 성능 파라미터 예

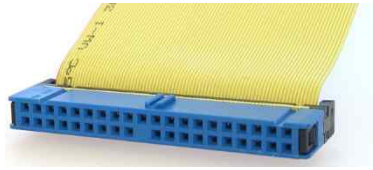
Characteristics	Seagate Barracuda ES.2	Seagate Barracuda 7200.10	Seagate Barracuda 7200.9	Seagate	Hitachi Microdrive
Application	High-capacity server	High-performance desktop	Entry-level desktop	Laptop	Handheld devices
Capacity	1 TB	750 GB	160 GB	120 GB	8 GB
Minimum track-to-track seek time	0.8 ms	0.3 ms	1.0 ms	—	1.0 ms
Average seek time	8.5 ms	3.6 ms	9.5 ms	12.5 ms	12 ms
Spindle speed	7200 rpm	7200 rpm	7200	5400 rpm	3600 rpm
Average rotational delay	4.16 ms	4.16 ms	4.17 ms	5.6 ms	8.33 ms
Maximum transfer rate	3 GB/s	300 MB/s	300 MB/s	150 MB/s	10 MB/s
Bytes per sector	512	512	512	512	512
Tracks per cylinder (number of platter surfaces)	8	8	2	8	2

2.7 디스크 성능 향상

- **하드 디스크의 성능향상 방법**
  - 디스크의 회전속도(RPM : Revolutions Per Minute) 향상
    - RPM이 높을수록, 발열, 소음, 고장에 취약.
  - 자기 기록 밀도를 높인다.
    - 용량과 전송속도 향상
  - 효율적인 섹터 배치
  - 디스크 Cache 사용
    - 디스크 내부에 Cache를 추가하여 디스크 접근속도를 향상.

2.8 HDD 버스 인터페이스

- **Parallel ATA (P-ATA) : 외부 저장장치 연결 표준 인터페이스**
  - ATA-1 (IDE) : 1986, 16-bit width, 16MBps bandwidth, 40-pin,
  - ATAPI : ATA 확장. 디스크 이외이 디바이스 연결.
  - ATA-2(EIDE) : 1994, ATA-1 확장.
  - ATA-4 (UDMA) : 80-bit, ~133MBps



- **Serial ATA (SATA)**
  - 2003, 8-pin, 1.5Gbps (SATA-1), 3Gbps (SATA-2), 6Gbps (SATA-3)

- **SCSI**
  - 1978, 68-, 80-pin,5MBps (SCSI-1) ~640MBps (Ultra-640)

- **SAS (Serial Attached SCSI)**
  - 3.0 Gbps, 6.0 Gbps

- **Fiber Channel**
  - 1988, 200MBps ~ 5100 MBps (20GFC)

RAID

1. RAID 개요

- **RAID(Redundant Array of Independent Disks)**
  - 저비용의 소용량 디스크를 연결하여 하나의 디스크처럼 사용하는 것.
    - 하드 디스크의 접근속도와 신뢰성 향상을 목적으로 제안.
  - 고속의 대용량 디스크대신 저비용의 소용량 디스크를 병렬로 배열하여 접근속도를 향상.
  - 다중-디스크 데이터베이스 설계를 위한 표준화된 기법.

- **디스크 병렬화 방법과 이점**
  - 여러 개의 디스크를 병렬로 사용하여 데이터를 분산저장 → 데이터 처리 성능 향상
  - 데이터의 중복 저장 및 에러 검사 데이터를 추가 저장 → 데이터 저장 안정성 및 신뢰성 향상

- **RAID 는 저장공간에 대한 신뢰도, 가용성, 접근속도, 용량을 향상**

1. RAID 개요

- **RAID 특징**
  - 여러 개의 물리적 디스크 드라이브들이 운영체제에 의해 하나의 논리적 드라이브로 취급.
  - 데이터를 스트라이핑(Striping) 방법으로 물리적 디스크들에 분산 저장.
    - 스트라이핑 : 파일을 여러 개의 조각으로 분할하여 분산 저장하는 방법.
  - 중복(Mirroring) 저장
    - 안전성 확보를 위해 데이터를 중복해서 분산 저장.
  - 디스크 오류발생시 데이터 복구를 위한 에러 코드를 함께 저장
    - 데이터 저장공간을 추가적으로 사용.

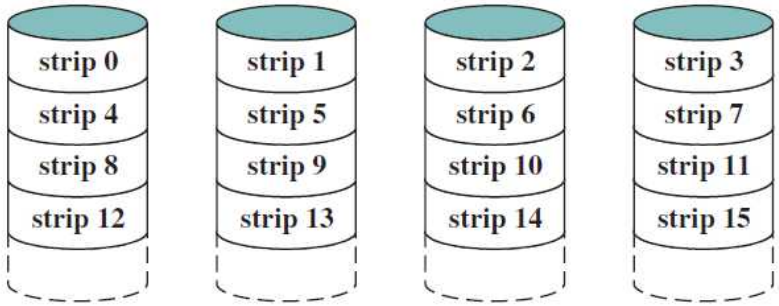
1. RAID 개요

- **RAID 레벨**
  - 표준 레벨 : 0~6 레벨
    - Hybrid RAID : 표준 레벨 조합. 1+0, 0+1, 5+0, N+N
    - 비표준 레벨 :

- **레벨 분류 기준**
  - 스트라이핑 : 파일을 스트립(Strip) 단위로 분할 및 분산 저장 방법.
  - 중복 저장(Mirroring) : 데이터 중복 저장
  - 에러 코드 : 에러 감지 코드 생성 및 저장 방법

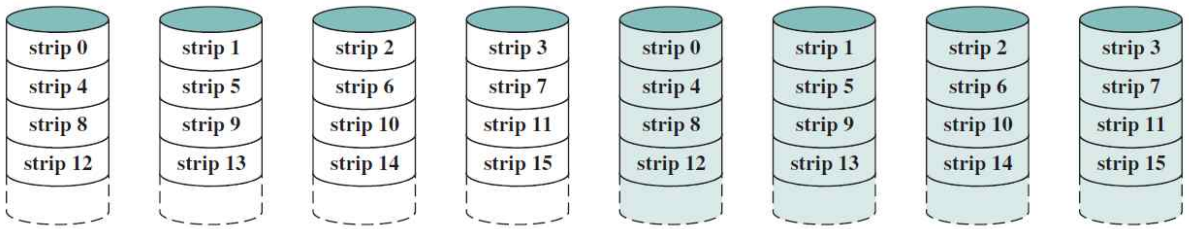
2. RAID 0

- **구성**
  - 스트라이핑 : 사용, Strip 단위로 저장, 병렬 읽기/쓰기 가능 → 접근 속도 개선
  - 중복 저장 (Mirroring) : 없음 → 가용성(Availability) 저하
  - 에러 코드 : 없음 → 오류 감지 불가
- **특징**
  - 1 개의 논리적 디스크를 여러 개의 물리적 디스크로 분산 저장
  - 라운드 로빈(Round Robin) 방식으로 스트립을 분산저장.
    - 한번의 request로 여러 strip을 동시에 처리할 수 있어서, 디스크 입출력 대기시간 절약
    - 빠른 데이터 접근이 필요한 응용분야에 적합
    - 디스크를 모두 데이터 저장용으로 사용하고, 추가 디스크 사용 없음.



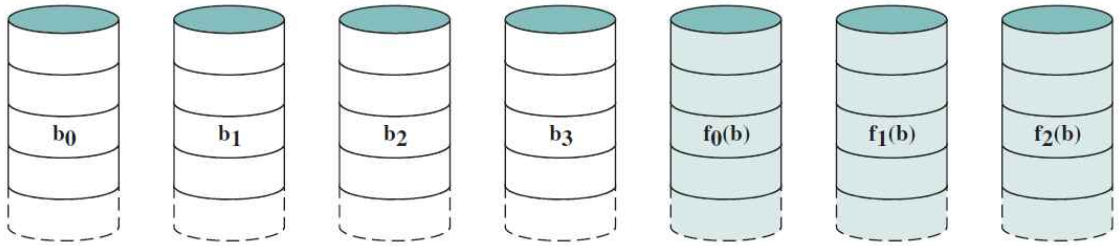
3. RAID 1

- **구성**
  - 스트라이핑 : 사용
  - 중복 저장(Mirroring) : 사용 : 데이터를 중복 저장
    - 읽기 : 2 개의 디스크 중 빠른 곳 선택
    - 쓰기 : 2 개의 디스크에 동시 쓰기, 쓰기 속도는 가장 느린 디스크에 의해 결정.
  - 에러 코드 : 없음
- **특징**
  - 1 개의 논리적 스트립을 2 개의 물리적 디스크에 복사 저장.
  - 디스크 장애 발생시 복구가 간단
    - 디스크 스왑(swap & re-mirror)후 다시 미러링, 복구시간 최소화.
  - 높은 가용성, 비용 부담 2배



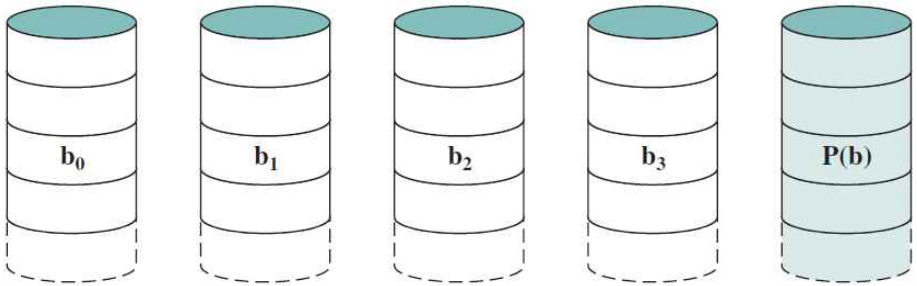
4. RAID 2

- **구성**
  - 스트라이핑 : 사용, Bit-level 스트라이핑
  - 중복 저장(Mirroring) : 없음.
  - 에러 코드 : 사용, Hamming Code 사용 (1-bit error 감지 및 정정, 2-bit error 감지)
- **특징**
  - 구현이 복잡하고, 고비용.
  - 오류 발생이 빈번한 환경에서 사용.
  - 병렬 액세스 기법 사용.
    - 디스크 회전 동기화 : 각 헤드는 디스크상에서 항상 동일한 위치를 유지



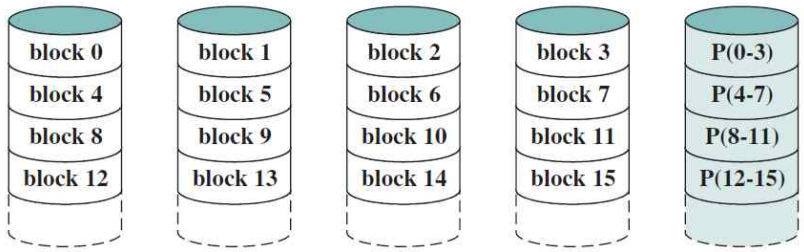
5. RAID 3

- **구성**
  - 스트라이핑 : 사용, Bit-level 스트라이핑
  - 중복 저장(Mirroring) : 없음.
  - 에러 코드 : 사용, 패리티 비트 1 개 사용 → 추가비용 적다.
- **특징**
  - RAID 2 와 유사
  - 디스크에 병렬 접근 → 높은 대역폭 응용에 적합



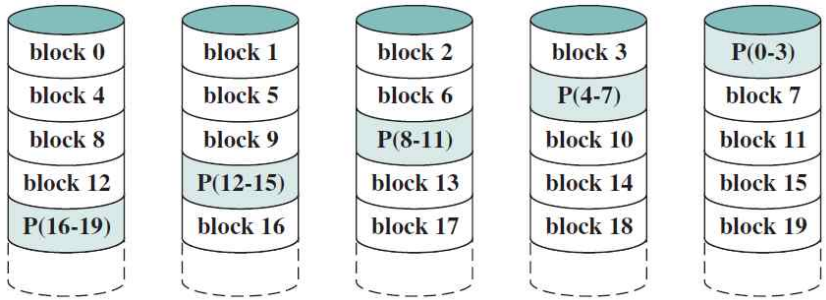
6. RAID 4

- **구성**
  - 스트라이핑 : 사용, 블럭단위 스트라이핑
  - 중복 저장(Mirroring) : 없음.
  - 에러 코드 : 사용, 패리티 비트 1 개 사용 → 추가비용 적다.
    - Bit-by-bit parity strip 저장.
- **특징**
  - RAID 3 와 유사
  - 독립 액세스 기법 사용 : 각 디스크는 독립적으로 동작
  - 많은 I/O 작업이 필요한 응용에 적합.
  - 높은 데이터 전송이 필요한 응용에는 부적합
  - 모든 쓰기 동작에 패리티 디스크 접근 : 패리티 디스크에 대한 병목(Bottle-neck) 현상 발생



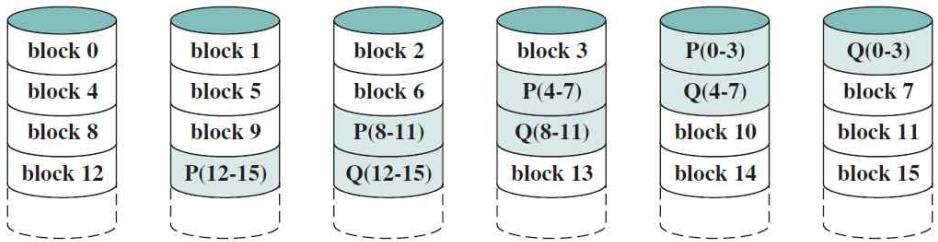
7. RAID 5

- **구성**
  - 스트라이핑 : 사용, 블럭단위 스트라이핑
  - 중복 저장(Mirroring) : 없음.
  - 에러 코드 : 사용, 패리티 분산 저장.
- **특징**
  - RAID 4 와 유사
  - 패리티 블록을 전체 디스크에 분산 저장
    - Round robin allocation
    - RAID 4 의 병목현상을 방지.
  - 트랜잭션 지향 응용에 적합
    - 네트워크 서버에서 많이 사용.
    - 가장 많이 사용하는 RAID 레벨.



8. RAID 6

- **구성**
  - 스트라이핑 : 사용, 블럭단위 스트라이핑
  - 중복 저장(Mirroring) : 없음.
  - 에러 코드 : 사용, 2중 패리티 분산 저장. 신뢰도 향상.
- **특징**
  - RAID 5 와 유사
  - 2개의 서로 다른 패리티 계산 후, 서로 다른 디스크들에 분산 저장
    - 서로 다른 2개의 알고리즘을 사용해서 parity bit 생성.
    - N+2 개의 디스크 필요, 2개 디스크 고장발생시에도 데이터 복구가능.
  - 쓰기동작에서 2 개의 패리티 블록 갱신 필요
    - 높은 쓰기 패널티(Write Penalty) 발생.



SSD

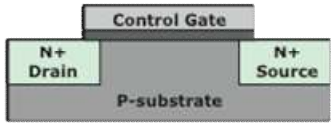
1. Solid State Drive

- **Solid State**
  - 반도체(semiconductor) 상에 구현된 전자회로를 의미.

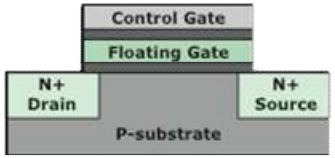
- **Solid State Drive (SSD)**
  - Solid State 부품 (플래시 메모리) 로 구현된 메모리 디바이스
  - 마그네틱을 사용하는 하드디스크를 대체할 수 있는 디바이스

2. 플래시 메모리

- **플래시 메모리(Flash Memory) 구성**
  - 기존 트랜지스터 (MOSFET) 구조에 floating gate 를 추가한 FET 를 메모리 셀로 사용
  - 게이트에 고전압을 걸어, 전자(electron)가 산화층(oxide layer 을 통과해서 floating gate 에 모이게 한다. 이 전자들은 전원공급이 중단되어도 상태를 유지하게 된다.

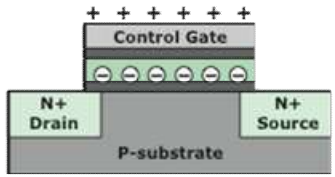


(a) Transistor structure



(b) Flash memory cell in one state

Normally high ('1') state



(c) Flash memory cell in zero state

3. 플래시 메모리 유형

- **플래시 메모리 유형 : NOR 타입, NAND 타입**

- **NOR**
  - 메모리 최소 접근단위
    - Read/write : byte 단위
    - Overwrite/erase : block 단위
  - Random access 가능
  - Fast read
    - 프로그램 저장에 사용 : 모바일 디바이스의 운영체제 코드 저장, 데스크탑 PC 의 BIOS 프로그램 저장용

- **NAND**
  - 메모리 최소 접근단위 : 16 or 32 bits
    - Read/write : page 단위 (random access 불가)
    - Overwrite/erase : block 단위
  - Slow read, fast write, small size(large capacity, low cost)
    - 대용량 데이터 저장에 사용 : USB flash drives, memory cards, SSDs

4. SSD와 HDD 비교

- **HDD와 비교했을 때 SSD의 장점**
  - 고속의 데이터 입출력 가능
    - access time 과 latency 가 짧음.
  - 내구성 : 물리적 충격이나 진동에 강함. 기계적 마모가 없음.
  - Low power consumption : 저 발열 (냉각장치 불필요)
  - No noise, Small size

	NAND Flash Drives	Disk Drives
I/O per second (sustained)	Read: 45,000 Write: 15,000	300
Throughput (MB/s)	Read: 200+ Write: 100+	up to 80
Random access time (ms)	0.1	4–10
Storage capacity	up to 256 GB	up to 4 TB

4. SSD와 HDD 비교

- **HDD와 비교했을 때 SSD의 단점**
  - Fragmentation에 의한 성능저하
    - SDD 는 블록 단위 write 를 하기 때문에, fragmentation 발생할 수록 성능이 저하된다.
      - 블록 크기 : 512KB, 128 pages/block, 4KB@page
    - 성능저하 방지 방법
      - 별도 여유공간을 마련 (over-provisioning)
      - 운영체제가 더 이상 사용하지 않는 블록에 대한 정보를 SDD에 통지해서 해당 블록을 삭제할 수 있게 한다. (TRIM 명령어 사용)

4. SSD와 HDD 비교

- **HDD와 비교했을 때 SSD의 단점(계속)**
  - SSD의 write 횟수 제한
    - 일반적인 최대 write 수 : 100,000
    - 수명 연장기술
      - wear-leveling : 쓰기 대상 블록을 균일하게 배분해서, 특정 블록에 write 가 집중되는 것을 방지.
      - 복잡한 bad-block 관리 기술 적용
      - 잔여 수명 예측 및 통보기능 : 시스템이 failure 대비가능

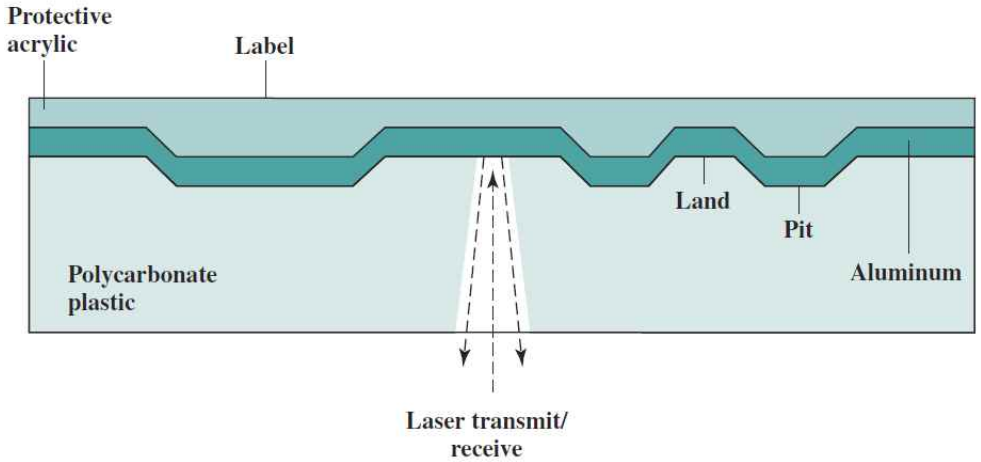
* 플래시 메모리에 page를 write하는 절차
① 쓰고자 하는 페이지를 포함하는 전체 블록을 RAM 버퍼에 복사한 후 내용을 수정한다.
② 플래시 메모리의 해당 블록을 erase 한다.
③ 수정한 블록을 플래시 메모리에 write 한다.

광학 디스크

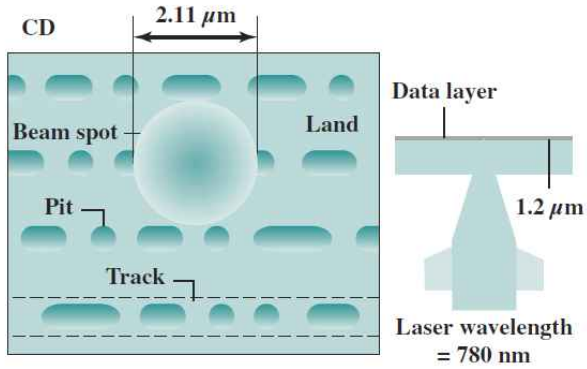
1. 광학 디스크

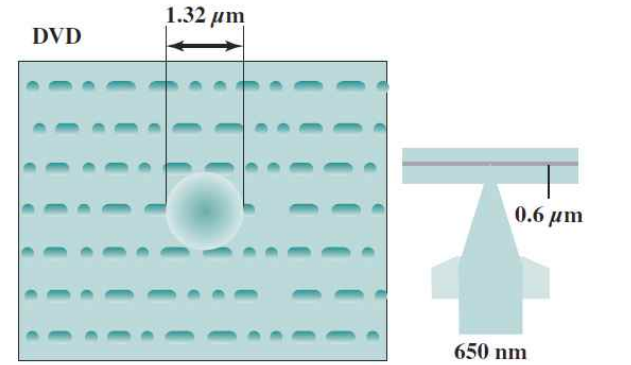
- 폴리카보네이트 (polycarbonate) 디스크상에 디지털 정보를 매우 작은 피트 (pit)로 인쇄한 후, 높은 반사율을 가진 알루미늄이나 금으로 코팅한 후, 먼지와 긁힘방지를 위해 아크릴 코팅한다.

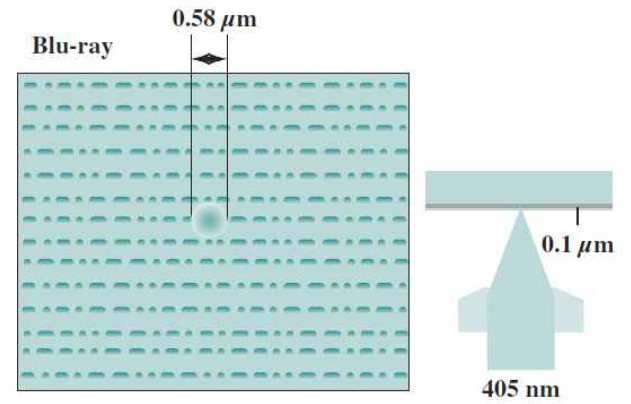
- 읽기 : 저전력 레이저를 사용하여 반사되는 레이저 빛의 강도의 변화를 감지한다.



2. 광학 디스크 특성

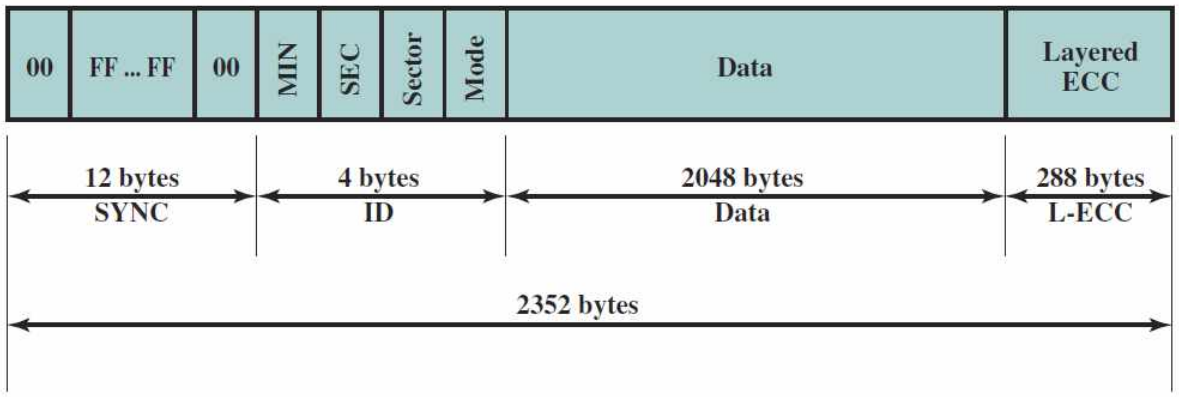

CD
2.11 μm
Beam spot
Land
Pit
Track
Data layer
1.2 μm
Laser wavelength
= 780 nm


DVD
1.32 μm
0.6 μm
650 nm


Blu-ray
0.58 μm
0.1 μm
405 nm

3. CD-ROM 포맷

- **SYNC 필드 : 블록의 시작**
- **헤더 (header) : 블록 주소 + 모드 바이트**
- **모드 바이트**
  - 모드 0 : 데이터 필드 비어있음.
  - 모드 1 : 2,048 Bytes Data + Error Correction Code
  - 모드 2 : 2,336 Bytes Data (No ECC)



4. 광학 디스크 용량

- **CD-ROM : ~650 MB**
- **DVD(Digital Versatile Disk) : ~17 GB**
- **Blue-ray (405nm blue-violet laser)**
  - ~50 GB/single-layer, ~100 GB/dual-layer
  - 36 Mbps (x1) ~ 432 Mbps (x12)

마그네틱 테잎

1. 마그네틱 테잎 저장장치

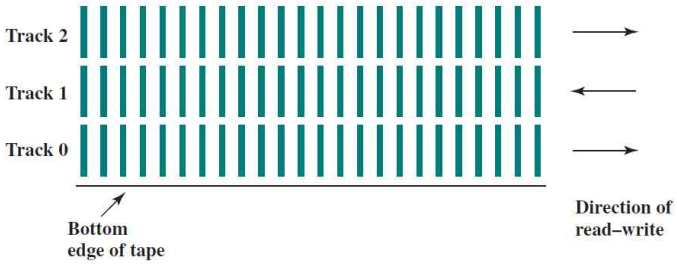
- **메모리 계층구조상 가장 저비용, 저속의 저장장치**

- **카트리지(Cartridge)방식 저장장치**

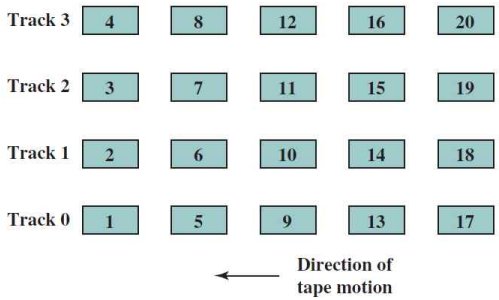
- **하드 디스크와 동일한 방법으로 읽기/쓰기**
  - 순차적 접근 방식
  - Tape width : 0.38(0.15")~1.27cm(0.5")
  - Tape 상의 트랙에 데이터 저장
    - 18~36 트랙 : 9-bit Grouping (8-bit Data + 1-bit Parity)

2. 마그네틱 테잎 레코딩

- **레코딩(Recording) 방법**
  - Serpentine 레코딩 : 트랙단위로 레코딩, 레코딩 방향 전환.



  - Block 레코딩 : 여러 트랙에 걸쳐 블록 단위로 레코딩.



3. 마그네틱 테잎 표준

- **LTO (Linear Tape Open)**
  - 1990 대에 개발된 개방형 표준

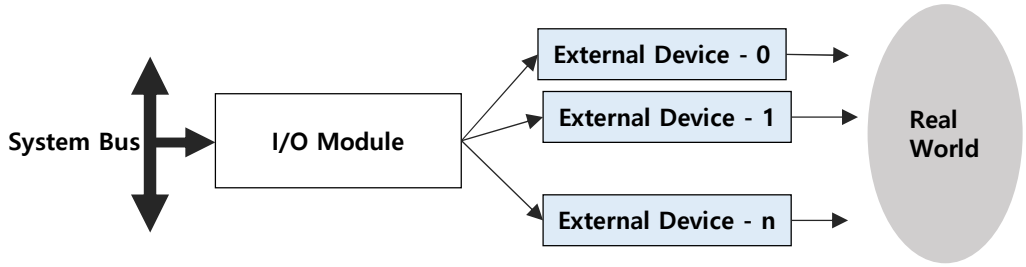
	LTO-1	LTO-2	LTO-3	LTO-4	LTO-5	LTO-6	LTO-7	LTO-8
Release date	2000	2003	2005	2007	2010	TBA	TBA	TBA
Compressed capacity	200 GB	400 GB	800 GB	1600 GB	3.2 TB	8 TB	16 TB	32 TB
Compressed transfer rate	40 MB/s	80 MB/s	160 MB/s	240 MB/s	280 MB/s	525 MB/s	788 MB/s	1.18 GB/s
Linear density (bits/mm)	4880	7398	9638	13250	15142			
Tape tracks	384	512	704	896	1280			
Tape length (m)	609	609	680	820	846			
Tape width (cm)	1.27	1.27	1.27	1.27	1.27			
Write elements	8	8	16	16	16			
WORM?	No	No	Yes	Yes	Yes	Yes	Yes	Yes
Encryption Capable?	No	No	No	Yes	Yes	Yes	Yes	Yes
Partitioning?	No	No	No	No	Yes	Yes	Yes	Yes

TBA : To Be Announced.

입출력 장치 개요

1. 외부장치

- **외부장치(External Device)는 컴퓨터와 컴퓨터 외부 환경사이의 데이터 입출력을 담당하는 장치**
  - 주변장치(peripheral), 입출력장치(i/o device)란 용어와 혼용
  - 외부장치는 I/O module을 통해서 컴퓨터 내부와 연결



1. 외부장치

- **외부장치 종류**
  - 사용자 인터페이스 장치
    - 컴퓨터 사용자와의 데이터 교환에 적합한 장치 : KVM(Keyboard, Video Display, Mouse)
  - 기계 인터페이스 장치
    - 기계장치와의 데이터 교환에 적합한 장치 : 센서(Sensor)와 구동장치(Actuator)
  - 통신장치
    - 원격에 있는 장치들과의 데이터 교환에 적합한 장치

1. 외부장치

- **외부 장치의 성능 척도**
  - 지연시간, 처리율, 고장 대처 능력

- **고장 대처 능력**
  - 신뢰성(Reliability)
    - 장치의 시간적 안정성. 주어진 조건에서 요청한 작업을 안정적으로 수행할 수 있는 능력.
    - 성능 척도 : 고장율, MTTF (Mean Time To Failure) 등
  - 유용성(Serviceability) : 장치 장애시 복구에 소요되는 시간.
    - 성능 척도 : MTTR
  - 가용성 (Availability)
    - 장치의 사용 가능성 척도.
    - 성능 척도 : 가동률 = (MTTF) / (MTTF + MTTR), MTTR(Mean Time To Repair)

- **장치 오류 발생시, 재시동, 결함 감내(Fault-Tolerance), 장애 원인 규명(Diagnosis) 등을 목적으로 장치 동작 상태 정보를 기록(Logging) 한다.**

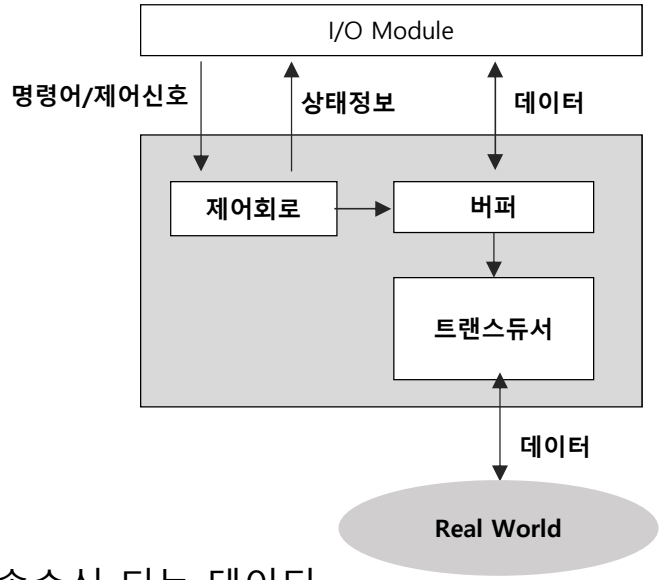
1.1 외부장치의 내부구조

- **트랜스듀서(Transducer)**
  - 전기적 형태의 데이터를 다른 에너지 형태로 변환하거나, 다른 형태의 에너지를 전기적 형태로 변환하는 회로, 센서(Sensor)

- **버퍼(Buffer)**
  - 데이터 임시저장 공간

- **제어회로(Control Logic)**
  - 외부장치에서 필요한 제어신호 발생

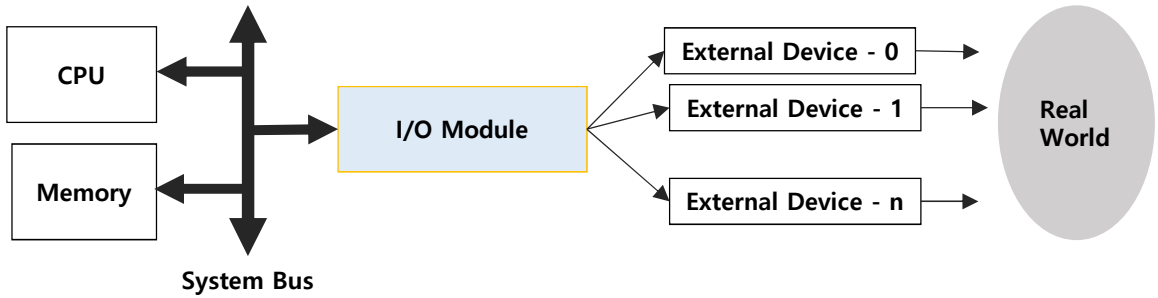
- **외부 인터페이스**
  - 명령어/제어신호 : 외부장치가 수행할 기능을 결정.
  - 상태정보 : 외부장치의 상태정보
  - 데이터 : I/O 모듈로 송수신되는 데이터, 외부세계로 송수신 되는 데이터



2. 입출력 모듈

- **입출력 모듈(I/O Moduel)은 외부장치를 컴퓨터 내부에 연결하는 장치를 말한다.**
  - 시스템 버스에 연결하여, 하나 또는 여러 개의 주변장치를 제어.
  - 시스템에 따라 I/O 채널(Channel), I/O 컨트롤러(Controller), Device Controller, I/O Processor로 불림.

- **I/O 모듈을 사용하는 이유**
  - 다양한 외부장치의 타이밍, 데이터 포맷, 제어방법 등을 단순화함으로써 CPU의 외부장치 처리부담을 경감.
  - 외부장치와 CPU/메모리의 처리속도 차이 극복.



2.1 입출력 모듈의 주요기능

- **제어 및 타이밍 신호 생성**
  - 내부 자원들과 외부장치들 사이의 통신흐름을 조정

- **CPU와의 인터페이스**
  - 명령 해독, 데이터 교환, 상태 보고, 어드레스 인식

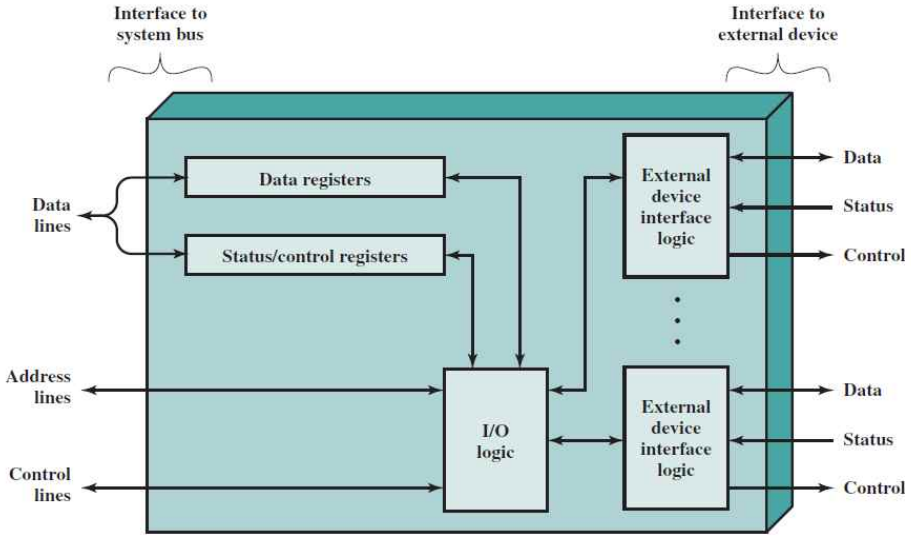
- **외부 장치와의 인터페이스**
  - 명령어, 상태 정보, 데이터

- **데이터 버퍼링(Buffering)**
  - 프로세서나 메모리와의 전송속도 불일치 해결하기 위해 데이터를 임시보관.

- **에러 검출(Error Detection)**
  - 오류를 검출해서 프로세서에 보고하는 기능

2.2 입출력 모듈의 내부구조

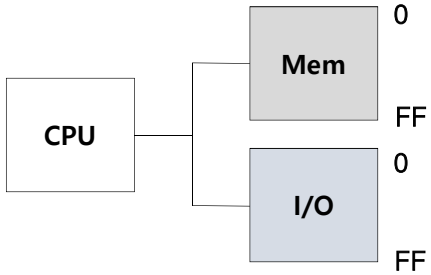
- **데이터 레지스터(Data Register)**
  - CPU와 송수신하는 데이터의 저장
- **상태 및 제어 레지스터(Status/Control Register)**
  - CPU에서 전달하는 명령어(제어정보)를 저장하거나, CPU에 전달할 I/O 모듈의 상태정보(각 외부장치의 상태정보)를 저장
- **제어회로(Control Logic)**
  - I/O 모듈 내부에서 필요한 제어신호 생성
- **외부 장치 인터페이스 회로**
  - 외부장치를 연결하는 회로
  - 외부장치와 데이터, 상태정보, 제어정보를 송수신
    - 여러 개의 외부장치를 연결가능



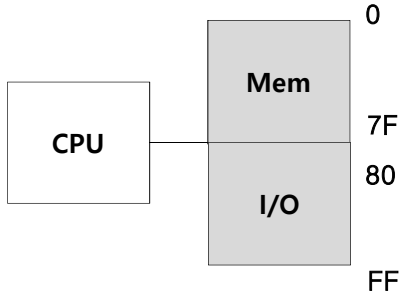
3. 입출력 장치의 어드레싱 방식

- **CPU는 어드레스를 사용하여 입출력 장치(I/O Device)를 액세스.**

- **I/O 장치에 어드레스를 할당하는 방식**
  - Memory-Mapped-I/O : 메모리와 I/O 장치가 동일한 주소 공간을 사용하는 방식
  - I/O-Mapped-I/O : 메모리와 I/O 장치가 별도 주소 공간을 사용하는 방식. Isolated-I/O.



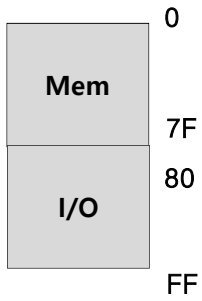
I/O-Mapped



Memory-Mapped

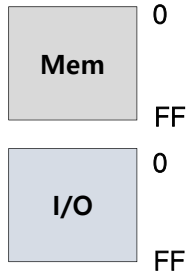
3.1 Memory-Mapped I/O

- **Memory-Mapped I/O**
  - I/O 장치와 메모리가 같은 주소공간을 사용
  - I/O 장치와 데이터 교환은 메모리 읽고 쓰기와 동일하게 처리
  - I/O 장치를 위한 별도 명령어가 불필요
  - 장점 : 다양한 메모리 접근 명령어들을 입출력 장치 접근시 사용가능
  - 단점 : 메모리 공간 낭비



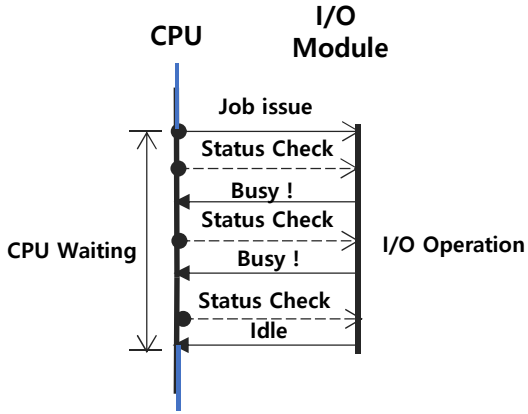
3.2 I/O-Mapped I/O

- **I/O-Mapped I/O**
  - I/O 장치와 메모리의 주소 공간을 별도 분리
  - I/O 장치와 메모리를 선택하는 제어신호가 필요
  - I/O 장치 전용 명령어 별도 필요
  - 장점 : 넓은 메모리 공간 사용
  - 단점 : 전용 명령어만 사용.



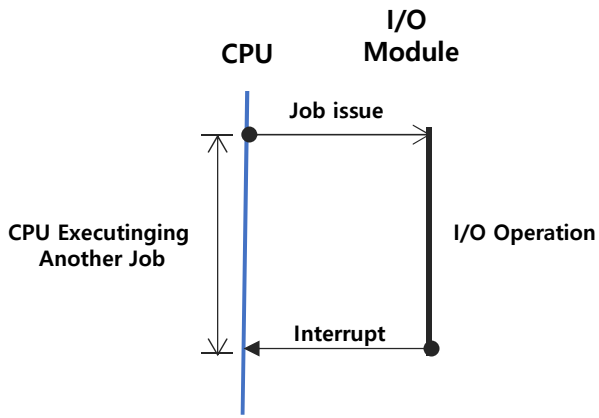
4. 입출력 장치의 데이터 전송 방식

- **Polled-I/O (Programmed-I/O)**
  - CPU가 입출력장치 동작을 직접 제어
  - CPU는 I/O 모듈이 작업을 끝낼 때까지 주기적으로 상태를 검사 (polling)
    - Busy : CPU Waitng (CPU 시간 낭비)
    - Idle : CPU는 다음 작업 지시.
  - 빈번하고 비교적 짧은 I/O 작업에 적합



4. 입출력 장치의 데이터 전송 방식

- **Interrupt driven-I/O**
  - Programmed-I/O 의 CPU 시간낭비 단점을 해결.
  - CPU는 I/O 모듈에 작업 지시후 다른 작업 수행. I/O 모듈이 작업수행.
    - CPU의 반복적 상태검사 작업 불필요
  - 작업완료 후 I/O 모듈이 인터럽트 발생.
  - 비교적 처리시간이 길고, 가끔씩 수행되는 I/O 작업에 적합.
    - 작업처리를 위한 상태 저장/복구 부담이 발생.



인터럽트

1. 인터럽트 개요

- **인터럽트 (Interrupt)**
  - CPU 동작이 다른 컴퓨터 구성요소 (메모리, 입출력장치) 에 의해 현재 실행중인 작업이 중지되는 것.
  - 프로세서의 성능향상을 위한 좋은 방법.
    - CPU 가 다른 구성요소의 작업이 완료될 때까지 기다리지 않고 다른 작업을 수행할 수 있다.
  - 인터럽트 상황이 발생 → 인터럽트 요청(Interrupt Request , IRQ)

- **인터럽트 처리방법**
  - 현재 실행중인 작업을 중지한 후, 인터럽트 서비스 루틴 (Interrupt Service Routine : ISR)을 사용하여 사전에 정의된 작업을 수행하고 복귀.

1.1 인터럽트 발생원

- **인터럽트 발생원 (Interrupt Source)은 프로그램 실행과정에서 발생하는 내부 인터럽트와 하드웨어에서 발생하는 외부 인터럽트로 구분할 수 있다.**

- **내부 인터럽트 발생원**
  - 실행시간 오류(Run-time Failure, Trap)
    - Divided by zero
    - 오버 플로우
    - 메모리 접근 위반(Violation)
  - Software Interrupt
    - SWI 명령어 실행
    - 디버깅 또는 명령어 확장용.

1.1 인터럽트 발생원

- **외부 인터럽트 발생원**
  - 타이머 인터럽트
    - 프로세서 내부의 타이머에서 발생.
    - pre-emptive, multi-tasking 에서 사용.
  - 입출력 인터럽트
    - 입출력 제어기 에서 요청
  - 하드웨어 동작 오류 (Hardware failure)
    - 메모리 패리티 (parity) 오류

1.2 인터럽트 감지 및 마스크

- **인터럽트 감지(Interrupt Detection)**
  - Polled 인터럽트 검사 : 주기적으로 인터럽트 플래그 검사.
  - Vectored 인터럽트 검사 : 인터럽트가 발생하면, 지정된 프로그램 메모리 번지로 자동으로 분기.

- **인터럽트 마스크(Interrupt Mask)**
  - 인터럽트 활성화 플래그 등을 사용하여 인터럽트 활성화 여부를 설정하는 것.
    - 마스크 불가능한 인터럽트 : NMI (Non-Maskable Interrupt)
      - Reset
  - 광역 마스크(Global Mask) : 모든 인터럽트에 발생원에 적용되는 마스크
  - 지역 마스크(Local Mask) : 특정 주변장치에만 적용되는 마스크

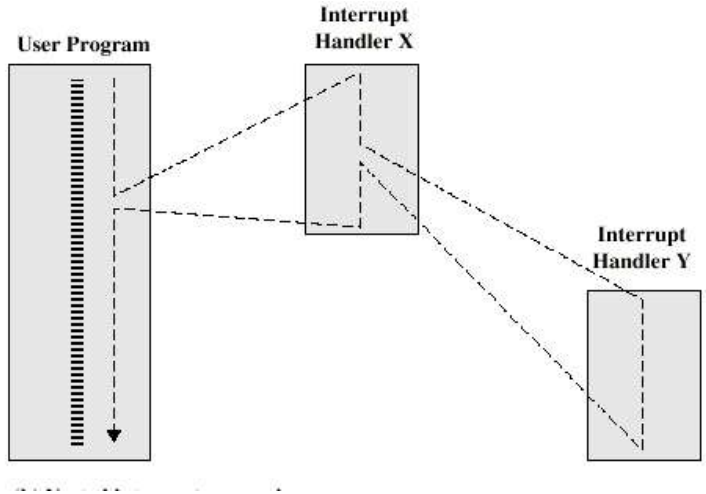
1.3 인터럽트 우선순위

- **인터럽트 우선순위(Interrupt Priority) 필요성**
  - 인터럽트가 동시에 발생했을 경우
    - 처리할 인터럽트를 결정
  - 인터럽트를 처리하는 도중에 다른 인터럽트가 발생할 경우
    - 추가로 발생한 인터럽트 처리여부 결정.

- **인터럽트 우선순위 설정**
  - 하드웨어적으로 우선순위 결정.

1.3 인터럽트 우선순위

- **인터럽트의 우선순위를 적용하여, 우선순위가 높은 인터럽트가 우선순위가 낮은 인터럽트보다 먼저 처리될 수 있도록 허용**
- **우선순위가 낮은 인터럽트의 처리시간 증가 가능성.**
  - 인터럽트 반응시간 지연
- **Nested 인터럽트 처리**



1.4 인터럽트 벡터

- **인터럽트 벡터 (Interrupt Vector)**
  - 인터럽트 발생시 자동으로 분기하는 어드레스
    - ISR 시작 어드레스
    - 하드웨어적으로 결정.

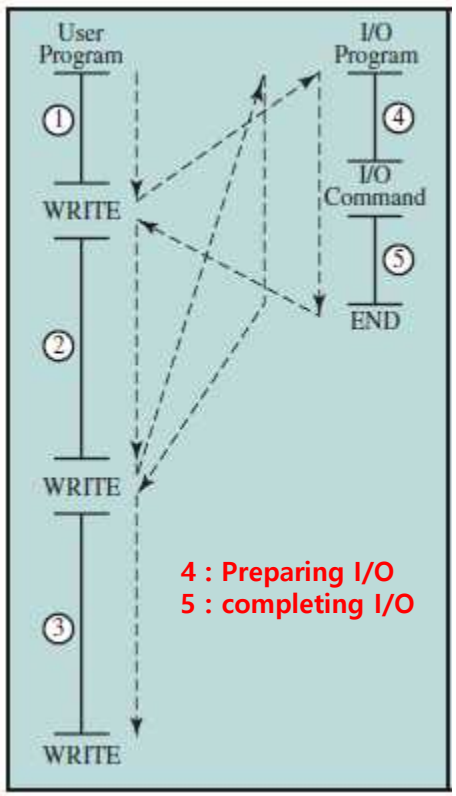
- **인터럽트 벡터 테이블 (Interrupt Vector Table)**
  - 인터럽트 벡터를 모아놓은 테이블
    - 대부분 메모리 첫 부분에 위치.

2. 인터럽트 서비스 루틴

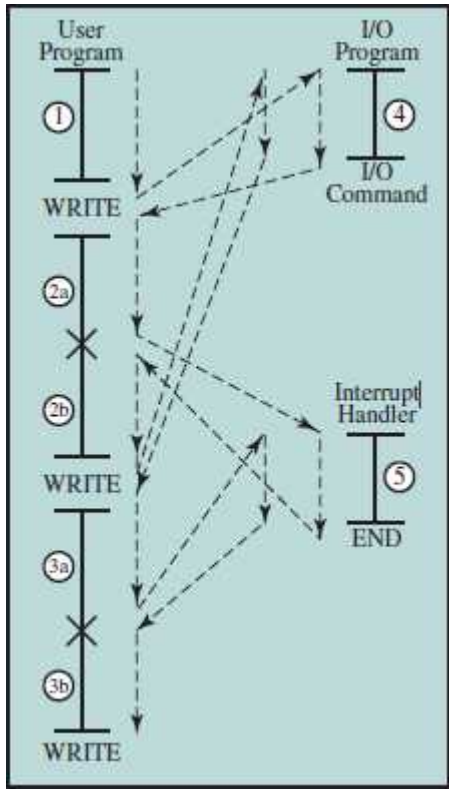
- **인터럽트 서비스 루틴(ISR : Interrupt Service Routine)**
  - 인터럽트를 처리하는 프로그램 코드.
    - 시스템에 의해 사전에 지정된 위치에 저장되어 있음.
  - 요청된 인터럽트를 처리하기 위한 사전에 정의된 작업을 수행한다.
  - 인터럽트 핸들러 (Interrupt Handler)라고도 함.

- **ISR 의 주요작업**
  - 다른 인터럽트의 중복 발생 방지.
  - CPU 상태정보 스택(Stack)에 저장
  - 사전에 정의된 작업 실행
  - 스택에 저장해두었던 CPU 상태를 복구
  - 다른 인터럽트 발생을 허용
  - 인터럽트 발생지점으로 복귀

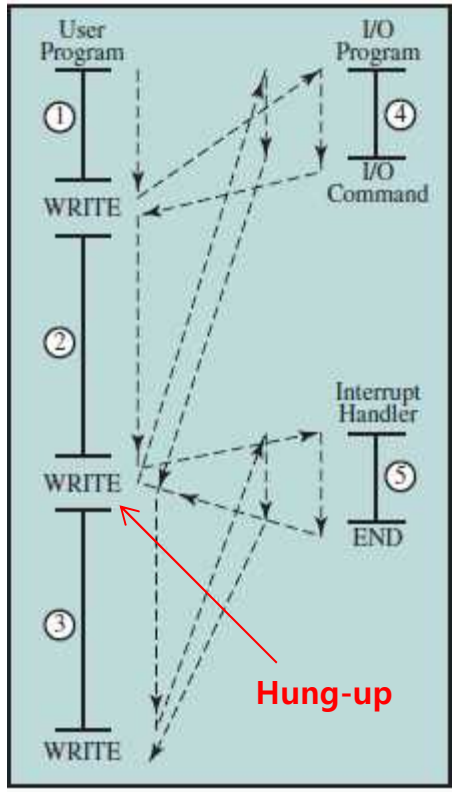
2.1 인터럽트 처리 흐름



No-Interrupt
(CPU Waiting)

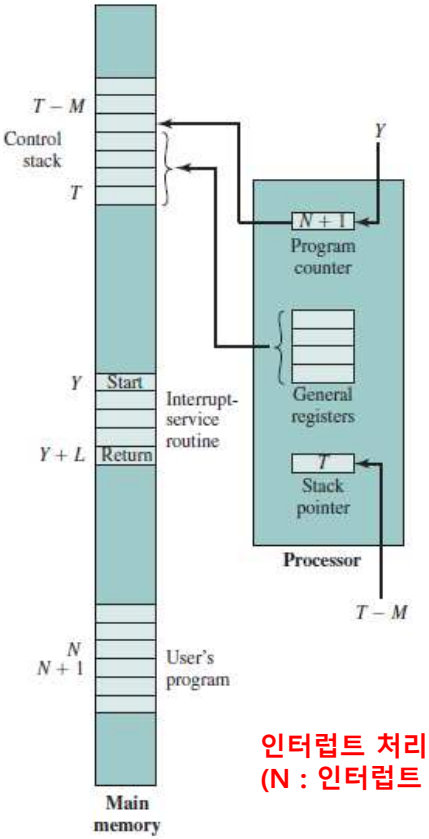


Interrupt
(Short I/O)

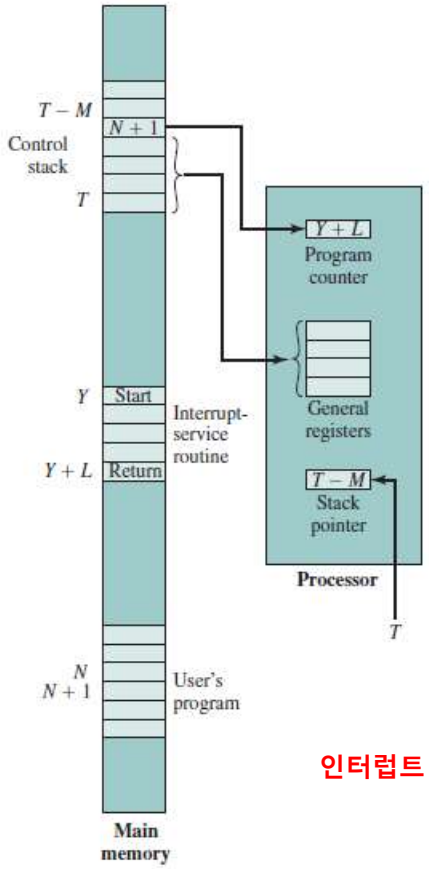


Interrupt
(Long I/O)

2.1 인터럽트 처리 흐름



인터럽트 처리전
(N : 인터럽트 발생시점)



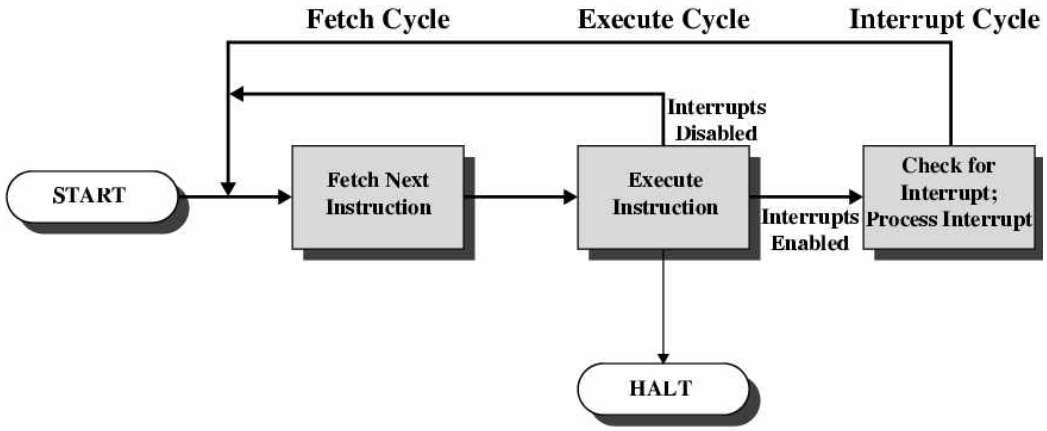
인터럽트 복귀시

3. 인터럽트 사이클

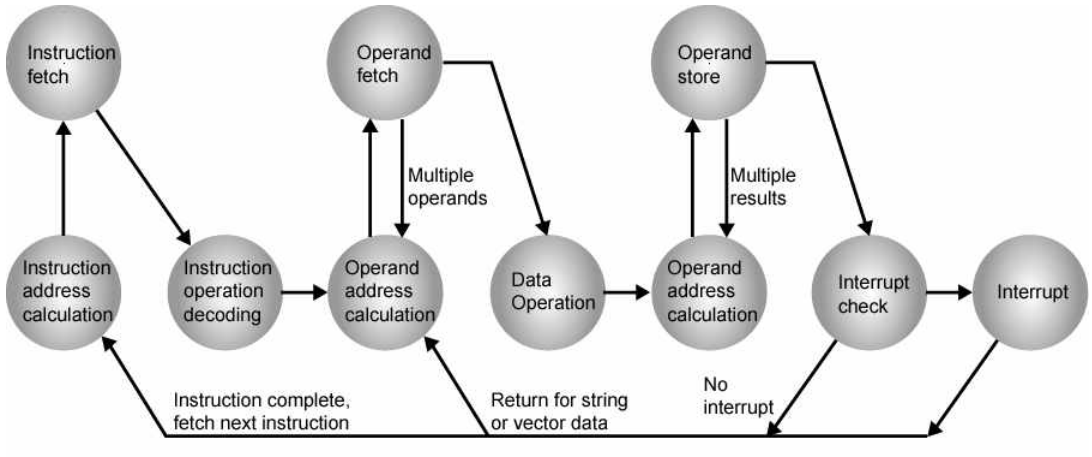
- **인터럽트 사이클 (Interrupt Cycle)**
  - 프로세서가 인터럽트 신호를 조사하여 인터럽트의 발생 여부를 검사하는 사이클
  - 명령어 사이클에 추가됨

- **프로세서는 인터럽트 발생 여부를 검사**
  - 대기 중인 인터럽트가 없으면, 다음 명령어를 인출.
  - 대기 중인 인터럽트가 있으면,
    - 현재 프로그램 실행을 중단
    - 컨텍스트(Context) 를 저장.
    - PC(Program Counter) 에 인터럽트 처리기(Interrupt Handler) 의 시작 주소 설정.
    - 인터럽트 처리
    - 저장한 컨텍스트 복구한 후, 인터럽트가 발생한 프로그램을 계속 실행.

3.1 인터럽트를 고려한 명령어 사이클



3.1 인터럽트를 고려한 명령어 사이클



4. 다중 인터럽트 처리

- **인터럽트가 동시에 여러 개가 발생하거나, 인터럽트 처리 중 다른 인터럽트가 발생했을 경우 처리방법은 실시간 처리성능을 결정하는 중요한 설계요소.**

- **다중 인터럽트 처리방법**
  - 다른 인터럽트 발생을 원천적으로 방지하는 방법
    - 나중에 발생한 인터럽트 무시.
  - 인터럽트 발생원에 우선순위를 적용하는 방법
    - 인터럽트의 우선순위를 적용하여, 우선순위가 높은 인터럽트가 우선순위가 낮은 인터럽트보다 먼저 처리될 수 있도록 허용
    - 우선순위가 낮은 인터럽트의 처리시간 증가 가능성.
      - 시스템의 실시간 처리능력 저하

4. 다중 인터럽트 처리

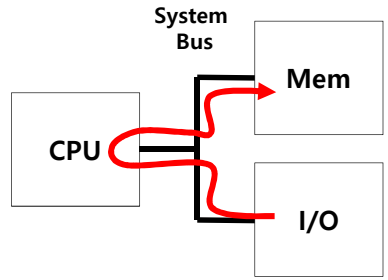
- **다중 인터럽트 처리방법(계속)**
  - 인터럽트 대기(Pending)
    - 인터럽트 처리동안 발생한 인터럽트는 대기 상태 (Pending) 로 있다가, CPU가 인터럽트 처리가능 상태가 되면, 인터럽트 처리.
    - 순차적 인터럽트 처리
      - 인터럽트 발생 순서에 따라 순차적으로 처리.
    - 단점
      - 상대적 우선순위나 시간적 긴급성을 고려하지 않음.

DMA

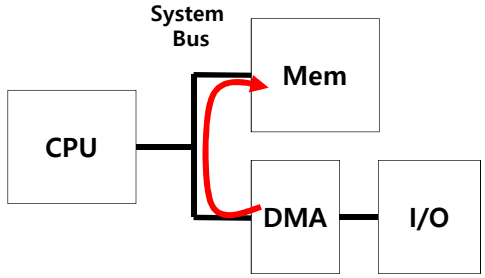
1. DMA 개요

- **기존 CPU를 경유한 입출력 장치와 메모리 사이의 데이터 전송의 문제점.**
  - CPU 개입으로 인한 데이터 전송속도 저하.
  - CPU 시간 낭비.
  - 해결방법 : 입출력 장치와 메모리의 직접 데이터 송수신.

- **DMA(Direct Memory Access)**
  - CPU를 경유하지 않고, 메모리와 입출력 장치와의 직접 데이터를 전송하는 방식
    - 시스템 버스에 DMA Controller 를 추가.
    - 데이터 전송에 CPU 개입 제거.
    - 블록단위 데이터 전송에 효과적



Polled/Interrupt-driven I/O

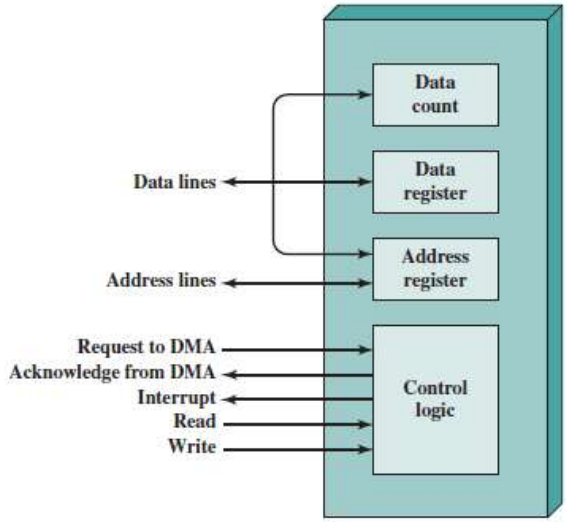


DMA

2. DMA 컨트롤러

- **CPU 인터페이스**
  - Data Lines : DMA를 통해 주고받을 데이터 통로
  - Address Lines : DMA 컨트롤러 어드레스.
  - Read/Write : 데이터 송수신 제어신호. Read(Input), Write(Output).
  - Interrupt : DMA 작업완료시 CPU에 전송되는 인터럽트 신호선.
  - Request to DMA : CPU의 DMA 작업요청 신호선.
  - Ack from DMA : DMA 컨트롤러의 통지 신호선.
    - CPU의 DMA 작업요청에 대한 응답.

- **내부 레지스터**
  - Data Count Register : DMA 동작에서 송수신할 데이터 량.
  - Data Register : 송수신 데이터의 임시 저장장소
  - Address Register : DMA 동작에 관련된 어드레스 저장장소
    - DMA에 참여하는 입출력 장치의 어드레스 또는 메모리 어드레스



3. DMA 동작

- **DMA 컨트롤러는 CPU로부터 제어권을 넘겨 받아, 프로세서 기능의 일부를 대신처리.**
  - CPU가 DMA 작업요청시 DMA 컨트롤러에 넘겨주는 정보
    - 작업유형 (Read/Write)
    - 디바이스 어드레스
    - 데이터 전송에 사용될 메모리 블록의 시작 어드레스
    - 전송할 데이터 양

- **DMA 컨트롤러의 시스템 버스 사용 타이밍**
  - CPU가 버스를 사용하지 않을 때.
  - CPU를 일시적으로(1 Bus Cycle동안) 멈추게 하고 버스를 사용 (Cycle Stealing)

3. DMA 동작

- **DMA 동작절차**
  - CPU 가 DMA 컨트롤러에 DMA 요청
  - CPU는 다른 작업 실행.
  - DMA 컨트롤러가 입출력 장치와 메모리사이의 데이터 전송작업 실행.
  - 데이터 전송 작업이 완료되면 DMA 컨트롤러가 인터럽트 발생.

버스 개요

1. 버스

- **버스(Bus)**
  - 배선의 집합 : 여러 개의 선 (line) 으로 구성
  - 하나 이상의 장치들을 연결하는 통신 경로.
  - 컴퓨터에서의 버스 : 컴퓨터 구성요소(CPU, 메모리, 입출력장치) 들을 상호연결

- **버스 마스터(Bus Master)**
  - 버스 슬레이브와 통신하기 위한 버스 사이클을 시작하며, 모든 버스 요청을 제어

- **버스 슬레이브(Bus Slave)**
  - 버스 마스터에 응답

- **버스의 한계성**
  - 여러 개의 장치가 공유하기 때문에 한번에 하나의 장치만 버스를 사용할 수 있다.
  - 버스에 대한 소유권 결정하기 위해, 버스 중재 필요.

1. 버스

- **버스 특징**
  - 공유 전송 매체
    - 여러 장치가 버스를 공유, 한 번에 한 장치만이 데이터 전송가능.
  - Broadcasting
    - 한 장치가 전송한 신호를 버스에 연결된 다른 모든 장치들이 수신 가능
  - 버스 충돌 (Bus Collision)
    - 버스에 연결된 두 개 이상의 장치가 동시에 신호를 전송하면 충돌 -> 버스 중재기 (Bus Arbiter) 필요.

2. 버스 유형

- **시스템 버스 (System Bus)**
  - CPU와 메모리를 연결하는 버스.
  - 길이가 짧고, 고속, 동기식 동작.

- **I/O 버스 (I/O Bus)**
  - 다양한 종류의 입출력 장치를 연결하는 버스.
  - 버스 길이가 길고, 저속, 비동식 동작

3. 버스 성능

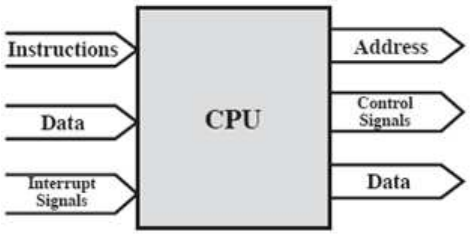
- **버스 대역폭(Bus Bandwidth)**
  - 대역폭 = 동작 주파수 x 버스 폭(Bus width)
  - Bus Throughput = bps(bit per second)

- **버스 대역폭 향상 방법**
  - 전용 버스 사용
  - 버스 폭의 확장
  - 블록 전송 적용
  - 동기화 버스 사용

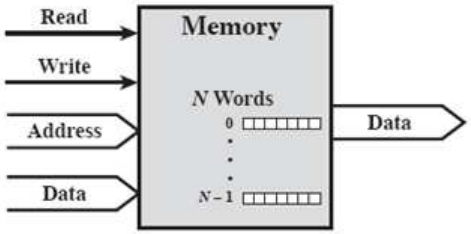
4. 컴퓨터 구성요소 상호연결

- **컴퓨터 구성요소에 따라 연결형태가 상이**
  - CPU 인터페이스 : 명령어, 주소, 데이터, 제어 신호, 인터럽트 신호
  - 메모리 인터페이스 : 주소, 데이터, 제어 신호
  - I/O 장치 인터페이스 : 주소, 데이터, 제어 신호, 인터럽트 신호

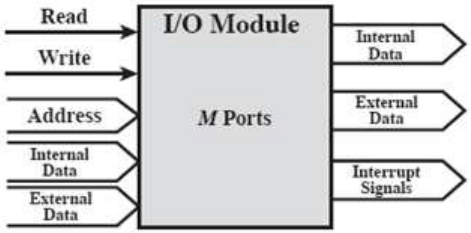
CPU의 입출력 신호



메모리의 입출력 신호

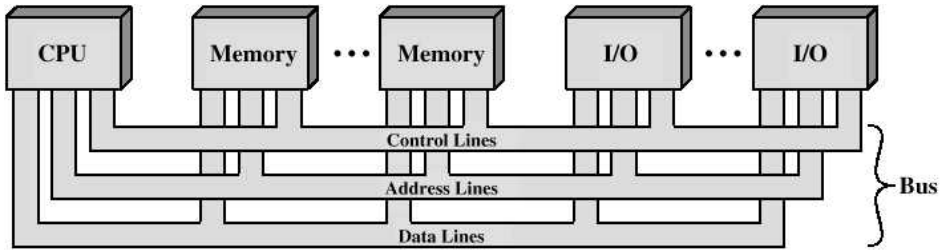


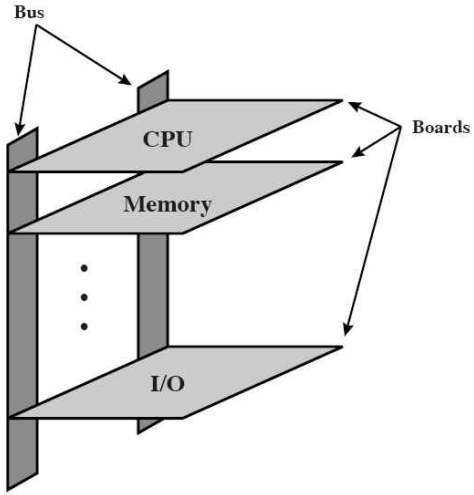
I/O 장치의 입출력 신호



4. 컴퓨터 구성요소 상호연결

- **컴퓨터 구성요소간의 데이터 전송 형태**
  - Memory to CPU
  - CPU to Memory
  - I/O to CPU
  - CPU to I/O
  - I/O from/to Memory : Direct Memory Access (DMA)
    - 프로세서를 통하지 않고 직접 연결





4.1 버스 신호선

- **데이터 선 (Data Line)**
  - 시스템 구성요소간의 데이터 전송

- **어드레스 선 (Address Line)**
  - 데이터의 근원지(Source) 나 목적지(Destination) 를 지정.

- **제어 선 (Control Line)**
  - 데이터 선과 어드레스 선의 사용을 제어하기 위해 사용
  - 시스템 구성요소들 사이의 명령어와 타이밍 정보를 전달.
    - 명령어 : 수행할 동작을 결정
    - 타이밍 정보 : 데이터와 어드레스 사이의 유효성 표시

4.2 제어 신호 선

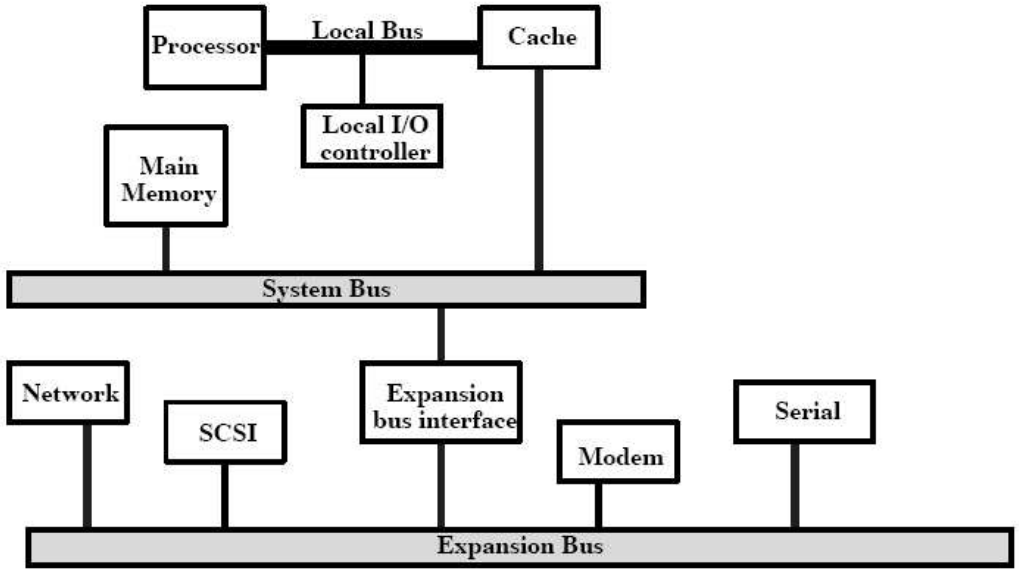
- **제어 및 타이밍 정보전달**
  - Memory read/write signal
  - I/O read/write signal
  - Transfer ACK : 데이터가 버스로부터 받아들여졌거나 버스상에 실렸다는 표시.
  - Bus request : 버스 사용 요청.
  - Bus grant : 버스 사용권 허가.
  - Interrupt request : 인터럽트 요청
  - Interrupt ACK : 인터럽트가 인식되었음을 통지.
  - Clock : 전송동작 동기화
  - Reset : 모듈 초기화

5. 다중 버스 계층구조

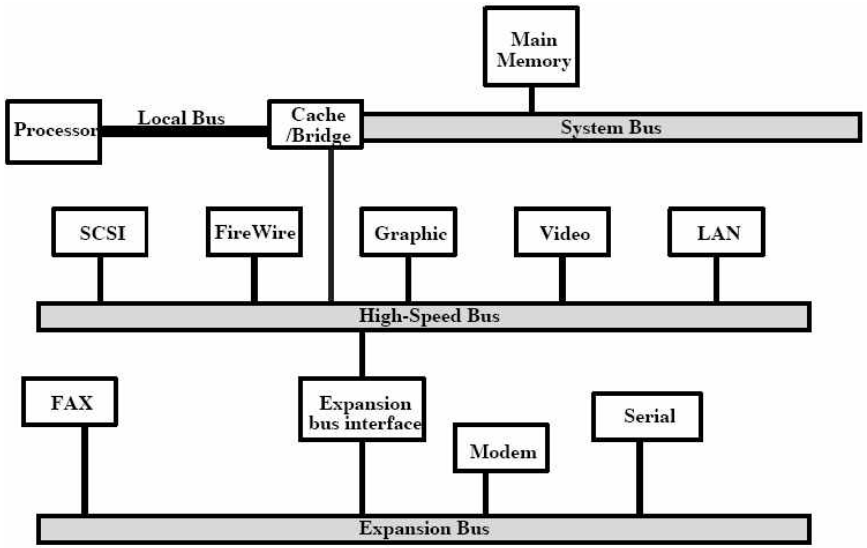
- **단일 버스에 아주 많은 장치들이 연결되어 있는 경우에 발생하는 문제**
  - 전파 지연(Propagation Delays)
    - 일반적으로 버스에 장치들이 더 많이 접속될 수록 전파지연이 길어진다.
    - 전파지연은 장치들이 버스 사용을 조정하는데 걸리는 시간을 결정.
  - 병목 현상(Bottleneck)
    - 전체 데이터 전송 요구가 버스 용량(Bus Capacity) 에 접근할 수록 버스는 병목이 된다.

- **대부분 시스템에서는 이 문제를 해결하기 위하여 다중 버스(Multiple Bus) 를 사용한다.**
  - 다중 버스 계층구조 (Multiple-Bus Hierarchy) 를 채용

5.1 다중 버스 계층구조 – ISA 버스 예



5.2 다중 버스 계층구조 – 고성능 버스 예



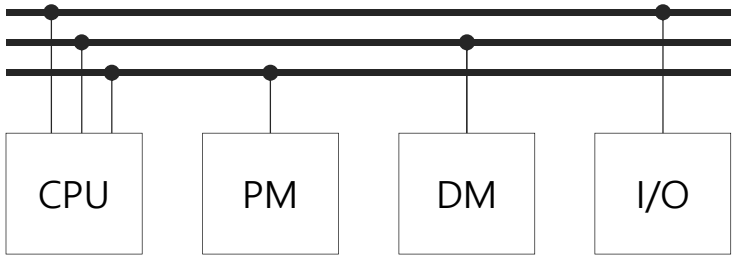
버스 설계

1. 버스 설계 요소

- **버스 설계의 기본 요소**
  - 버스 유형
    - Dedicated, Multiplexed
  - 중재 (Arbitration) 방법
    - Centralized, distributed
  - 타이밍 (Timing)
    - Synchronous, asynchronous
  - 버스 폭 (Bus Width)
    - Address, data
  - 데이터 전송 유형 (Data Transfer Type)
    - Read, write, read-modify-write, read-after-write, block

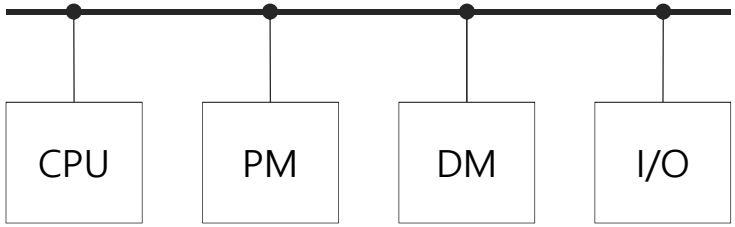
2. 버스 유형

- **전용 버스(Dedicated Bus)**
  - 용도가 지정되어 특정 용도로만 사용하는 버스
    - 데이터, 어드레스
    - 명령어, 데이터, 입출력 장치 연결 버스.
  - 장점 : 높은 데이터 처리율 (high-throughput)
    - 큰 대역폭 확보 가능
    - 여러 개의 장치가 동시에 데이터 전송가능.
  - 단점 : 시스템의 크기와 가격이 증가.



2. 버스 유형

- **다중화 버스 (Multiplexed Bus)**
  - 하나의 버스로 여러 가지 용도로 사용하는 버스
    - 시분할(Time-Multiplexed) 방식 적용
  - 제어 신호를 사용하여, 버스 용도를 결정.
    - 어드레스 유효, 데이터 유효 제어선 사용
  - 장점 : 버스 수가 적어 비용과 공간을 절약
  - 단점 : 복잡한 제어회로, 병렬처리가 어려워 성능저하 발생.



3. 버스 중재

- **여러 개의 장치가 하나의 버스를 공유하기 때문에 버스 중재(Bus Arbitration)가 필요.**
  - 버스 제어기(Controller) 또는 버스 중재기(Arbiter)를 사용하여 중재
  - 하나 이상의 모듈들이 버스를 제어할 수 있어야 한다.
    - CPU 및 DMA 제어기
  - 한번에 하나의 모듈만이 버스를 제어할 수 있다.

- **중재 방법**
  - 중앙 집중식 (Centralized)
    - 버스 제어기 또는 중재기가 버스의 사용시간을 할당.
  - 분산식 (Distributed)
    - 중앙 제어기 없이 각 모듈에 액세스 제어회로가 포함되어 있으며, 모듈들이 버스를 공유한다.

4. 버스 타이밍

- **버스 타이밍(Timing)**
  - 버스상에서 변화하는 신호(데이터, 주소, 제어신호)의 시간적 흐름

- **버스 타이밍 차트(Chart)**
  - 신호가 발생/변화되는 시점을 순서적으로 보여준다.

- **타이밍 유형**
  - 동기식 (Synchronous) : 클럭 기반 (Clock-based)
  - 비동기식 (Asynchronous) : 이벤트 기반 (Event-based)

4.1 동기식 버스

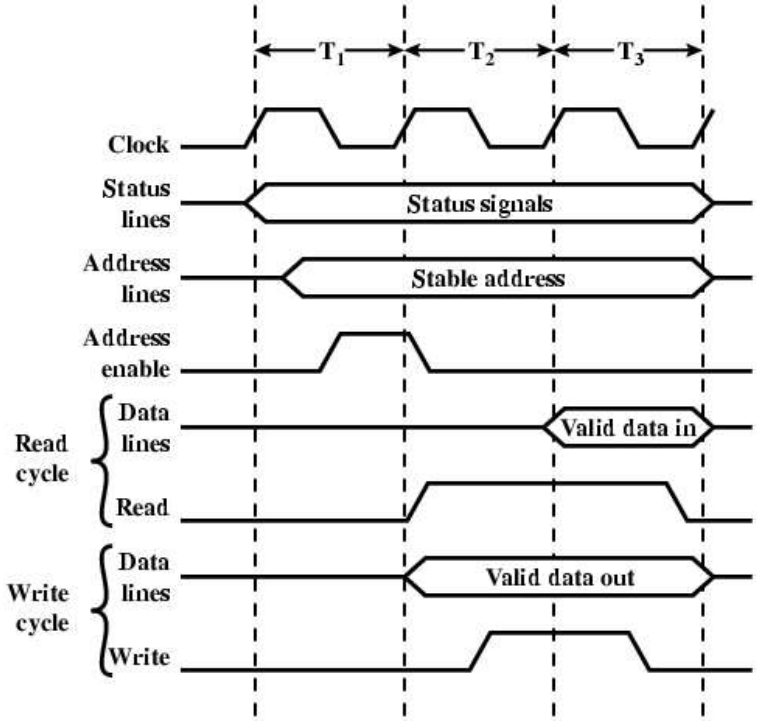
- **동기식 타이밍 (Synchronous Timing)**
  - 버스상의 이벤트 발생이 클록신호에 의해 결정된다.
    - 클록 싸이클 / 버스 싸이클 : 1-0 을 한 번 전송하는 것. 한 개의 시간 슬롯 (Time Slot) 에 해당.
  - 제어 버스에 클록 신호선이 포함.
    - 클럭 왜곡(Clock Skew) 줄이기 위해, 짧은 거리에 적용.

- **장점**
  - 고속 데이터 전송 가능
  - 구현과 검증이 쉽다.

- **단점**
  - 비동기식에 비해 융통성이 떨어진다.
  - 고정된 클럭 속도에 의해 제어되기 때문에 장치의 성능이 향상되어도 그에 따른 성능상의 이득을 얻기 힘들다.

4.1 동기식 버스

- 동기식 버스 타이밍 챠트



4.2 비동기식 버스

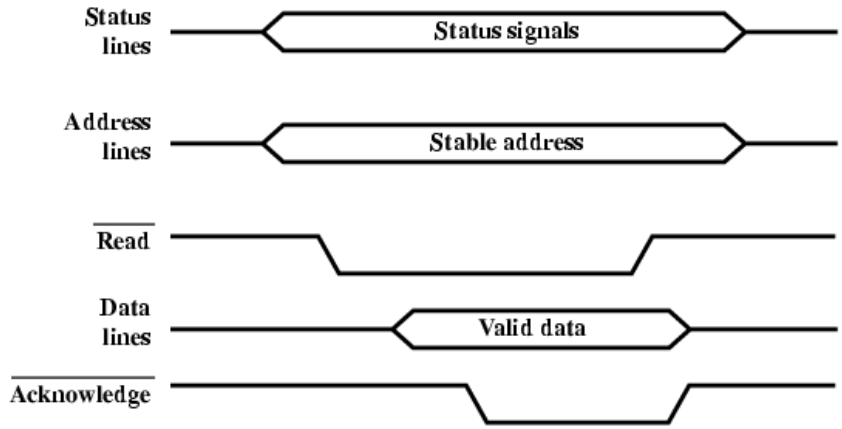
- **비동기식 타이밍 (Asynchronous Timing)**
  - 버스상의 이벤트 발생이 이전 이벤트의 발생에 의해 결정된다.
  - 데이터 전송은 송신기와 수신기 사이의 이벤트 신호 교환에 맞춰 이루어 진다.
    - Handshaking 방식.

- **장점**
  - 다양한 종류의 장치를 수용할 수 있기 때문에, 융통성이 높다.
    - 저속과 고속 장치를 같은 버스를 공유해도 성능저하가 없다.
  - 버스를 길게 할 수 있어, 많은 수의 장치를 연결할 수 있다.

- **단점**
  - 제어신호가 추가적으로 필요하기 때문에, 구현이 복잡하고, 속도가 느리다.
    - 저속 장치 연결에 적합.
  - 제어선의 잡음에 민감.

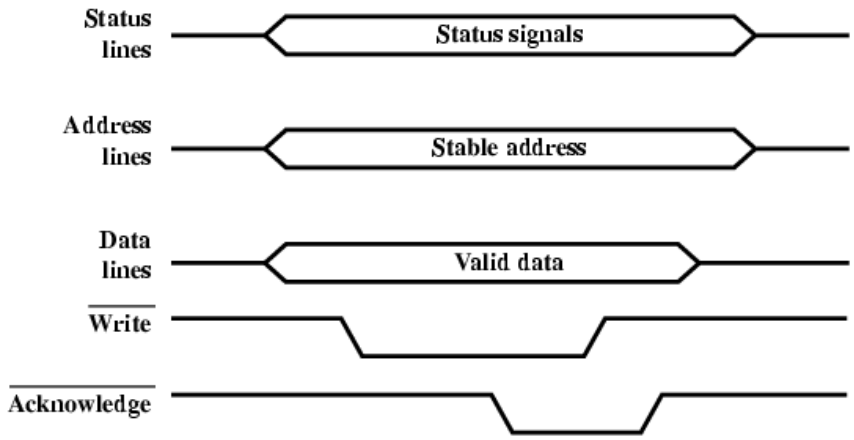
4.2 비동기식 버스

- 비동기식 타이밍 챠트 - 읽기



4.2 비동기식 버스

- 비동기식 타이밍 챠트 - 쓰기

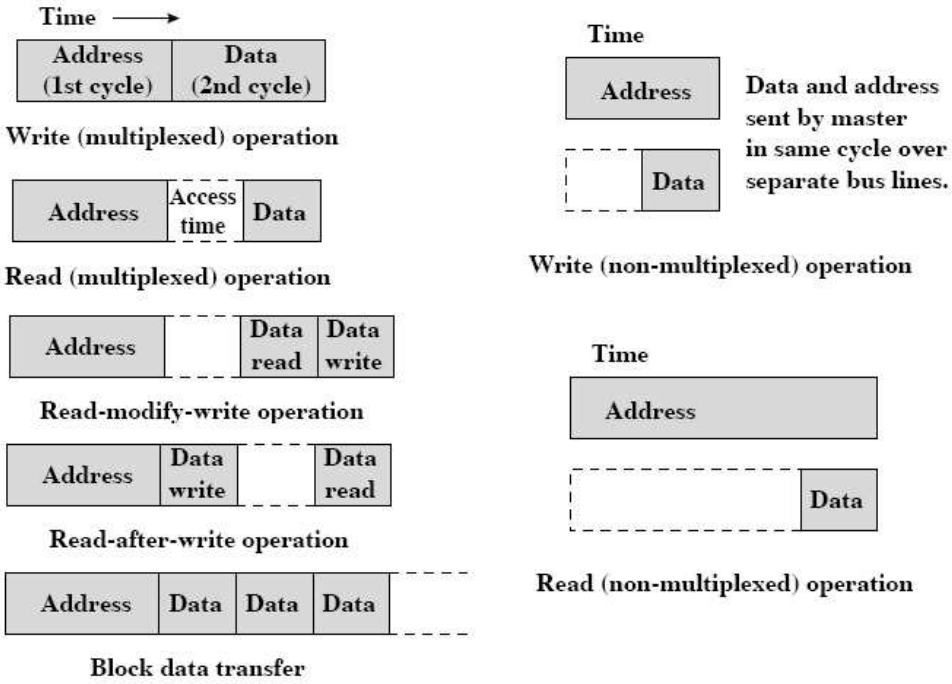


5. 버스 폭

- **데이터 버스**
  - 데이터 버스의 폭은 시스템 성능과 관계가 있다.
  - 데이터 버스 폭이 넓을수록, 한번에 전송할 수 있는 비트 수가 증가.

- **어드레스 버스**
  - 어드레스 버스 폭은 메모리 용량에 영향.
  - 주소 버스 폭이 넓을수록 지정할 수 있는 메모리 장소들의 범위가 증가.

6. 데이터 전송 유형


Time
Address (1st cycle)
Data (2nd cycle)
Write (multiplexed) operation
Address
Access time
Data
Read (multiplexed) operation
Address
Data read
Data write
Read-modify-write operation
Address
Data write
Data read
Read-after-write operation
Address
Data
Data
Data
Block data transfer
Time
Address
Data
Data and address sent by master in same cycle over separate bus lines.
Write (non-multiplexed) operation
Time
Address
Data
Read (non-multiplexed) operation

I/O 버스 표준

1. ISA

- **ISA 버스**
  - 8-bit @ 4.77MHz : 4.77 MB/s
  - 16-bit @ 8.33MHz : 16.67 MB/s

- **EISA**
  - 8-,16-,32-bit @ 8.33MHz : 266.56 Mbps

- **AGP (Accelerated Graphics Port)**
  - 컴퓨터 그래픽 카드용 버스, PCI-Express 로 대체
  - AGP 1.0 1x : 32-bit @ 66MHz, 266 MB/s
  - AGP 2.0 4x : 32-bit @ 66MHz, 1.066 GB/s
  - AGP 3.0 8x : 32-bit @ 66MHz, 2.133 GB/s

2. PCI

- **PCI (Peripheral Component Interconnection)**
  - 인텔에서 개발한 후, 일반 공개된 버스 규격.
  - 단일, 복수의 프로세서 기반 시스템용.
  - 동기식 버스. 주소와 데이터를 시분할 방식으로 전송.
  - 중앙 집중식 버스 중재 구조.
  - Rev. 2.0 : 32-bit, 64-bit @ 33MHz
  - Rev. 2.1 : 32-bit, 64-bit @ 66MHz
  - Max. rate : 528 Mbyte/sec (4.224 Gbps)
  - 최대 입출력 핀 : 100 pins
    - 49 개 핀은 필수
    - 51 개 핀은 선택

3. SATA

- **PATA (Parallel ATA)**
  - ATA (Advanced Technology Attachment) 의 다른 이름. IDE
  - 디스크와 같은 대용량 저장장치용 버스 규격, 병렬전송.
  - 16-bit, 133 MB/s, 병렬전송
  - 40-, 80-pin (Ribbon Cable), ~46.7 cm, SATA로 대체.

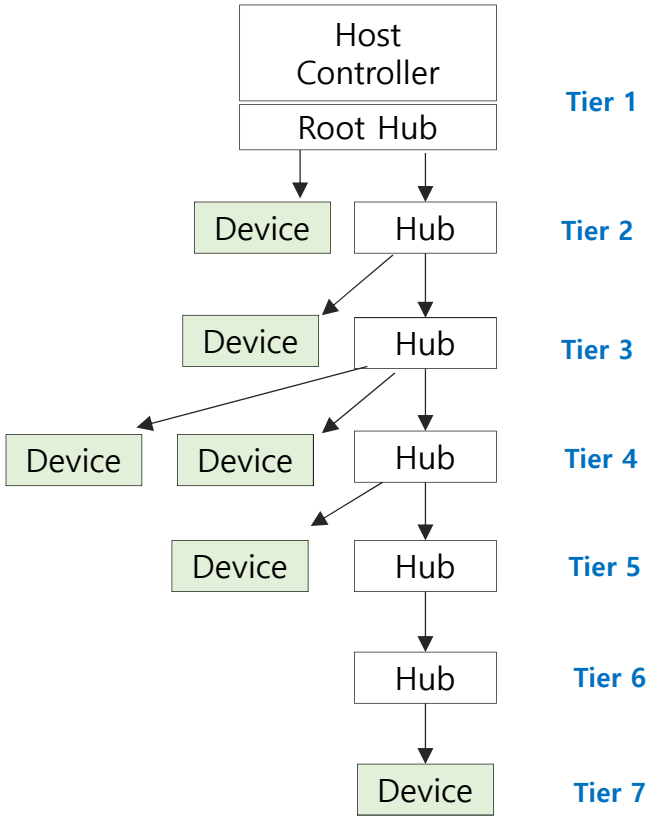
- **SATA (Serial ATA)**
  - 직렬전송, 8b/10b 인코딩 사용.
  - 7-pin, ~1m, serial, point-to-point
  - SATA 1.0 : 150 MB/s
  - SATA 2.0 : 300 MB/s
  - SATA 3.0 : 600 MB/s
  - SATA 3.2 : 1.97 GB/s

4. IEEE 1394

- **IEEE 1394**
  - 홈네트워크용 버스 규격. 멀티미디어 장치의 데이터 전송
  - FireWire(Apple), i.Link(Sony)
  - 4~5m, ~63 Port 지원, PnP 가능
  - IEEE 1394 400 : 49 MB/s
  - IEEE 1394 800 : 98 MB/s
  - IEEE 1394 3200 : 393 MB/s

5. USB

- **USB(Universal Serial Bus)**
  - 직렬전송, Tiered-Star 토폴로지
    - 1 호스트 컨토롤러에 최대 127 디바이스 연결가능
    - 루트 허브 : 호스트 컨트롤러에 내장된 허브, 2단자만 제공.
    - USB 허브는 5-tier 까지 추가 가능
  - ~5m
  - USB 1.1 : 1.5 MB/s
  - USB 2.0 : 35 MB/s
  - USB 3.0 : 400 MB/s



end