**컴퓨터 시스템의 기본 구조**

컴퓨터 시스템은 기본적으로 하드웨어와 소프트웨어로 나뉜다.

* 하드웨어 : 전자회로와 그 밖의 물리적인 장치들
* 소프트웨어 : 이를 활용할 수 있는 기본적인 프로그램이나 이에 따르는 기술들

**1.1 컴퓨터의 외부 구조**

(Vavr Vemory) 
(Varn Board) 
Ica") 
(Monitor) 
f/e/t•oard) 
3.21 
(MODEM. LAN card) 
n 
(Pr.nter) 
(Video Card) 
(CD-ROM) 
(Hard DISK) 

컴퓨터 = 본체 + 주변장치들 이 복합적으로 연결된 집합체.

* 본체 = 케이스 + 내부의 여러 장치들.
* 주변 장치 = 키보드 + 모니터 + 프린터 등.

**1.2 컴퓨터의 내부 구조**

중양처리 장치(CPU) 
제어 장치 
(Control Unit) 
연산 장치 
(ALU) 
입력 장치 
주기억 장치(RAM, ROM) 
보조기억 장치 
[그림 3.3] 컴퓨터의 내부 구조 
출력 장치 
••-> 자료 신호 
- 영령 신호 

기본적으로 입력장치, 중앙처리장치(Central Processing Unit: CPU), 주기억장치(Main Memory: MM), 보조기억장치(Auxiliary Memory), 출력장치 등으로 이루어져 있다.

내부적으로 중앙처리장치가 가장 중요한 역할을 함.

중앙처리장치(CPU) = 연산장치 + 제어장치

* 연산장치(Arithmetic&Logic Unit: ALU) = 사칙연산 및 논리 연산. 프로그램의 명령을 실행하기 위한 장치.(자료의 비교, 판단, 이동, 편집 수행)
* 제어장치(Control Unit) = 프로그램의 명령을 해석 + 명령에 따라 각 장치의 작업 지시 및 감독. 각종 입출력 장치, 기억 장치, 연산 장치 등의 동작을 제어하며, 주기억장치에 저장된 프로그램 해독 및 해독된 명령들을 각 장치에 보내 처리하도록 지시.

**1.3 컴퓨터 시스템의 구성 요소**

컴퓨터 시스템 = **하드웨어** + **소프트웨어** (+ 펌웨어(Firmware))

* **소프트웨어** = **시스템 소프트웨어**(System Software) + **응용 소프트웨어**(Application Software). (용도, 구성, 사용목적에 따라 구분)
  + **시스템 소프트웨어** : 컴퓨터를 효율적으로 운영하고 제어하기 위한 프로그램. 운영체제(Operating System) 등과 같은 소프트웨어.
  + **응용 소프트웨어** : 특정 분야의 업무를 처리하는 데 사용되는 소프트웨어.

(1) 하드웨어(hardware) = 컴퓨터를 구성하는 물리적인 장치들의 집합체

대표적인 하드웨어인 본체 = 메인 보드(main board) + 주변기기 로 구성됨.

메인보드 = 중앙처리장치(CPU), 롬(ROM), 램(RAM) 등과 같은 핵심 부품을 장착함.

주변기기 = 확장카드들, 보조기억장치(디스크 드라이브 등)

전원 공급장치(부품에 전기 공급), 입력장치, 출력장치 도 하드웨어에 속함

* 입력 장치 : 문자나 기호와 같은 데이터를 컴퓨터가 이해할 수 있도록 전기적인 신호로 변환. (키보드, 마우스, 통신 포트, 센서 등)
* 출력 장치 : 중앙처리 장치에서 처리된 결과를 출력. (모니터, 스피커, 프린터, 플로터 등)
* 중앙처리 장치 : 제어 장치, 연산 장치, 레지스터(register) 등으로 구성. 개인용 컴퓨터에선 마이크로프로세서라고도 부름.(마이크로프로세서의 이름은 속도에 따라 80286, 80386, 80486, Pentium I II III IV 등의 이름 사용)
* 주기억 장치 : 컴퓨터 내에서 명령어와 데이터들을 기억하는 대규모 저장 장치. 입력 장치로부터 들엉온 자료는 주기억 장치에 저장됨.
* 보조기억 장치 : 주기억 장치를 보조하는 기억 장치. 자기 디스크, 자기 테이프, 디스켓, 하드 디스크, CD-ROM, DVD, USB 등이 있다.

(2) 시스템 소프트웨어

사용자가 편하게 사용할 수 있도록 환경 조성, 사용자가 컴퓨터와 대화할 수 있는 user interface 제공.

**운영체제**, **언어 번역 프로그램**, **유틸리티(utility)** 등이 있다.

* **운영체제** : 컴퓨터의 전체적인 작동을 관리하는 프로그램. 편하고 효율적으로 사용하는 방법 제공.
* **언어 번역 프로그램** : 프로그래밍 언어로 작성된 프로그램을 컴퓨터가 이해할 수 있는 기계어로 번역하는 프로그램.
  + 어셈블리 프로그램을 번역 -> 어셈블러(Assembler)
  + 고급 언어를 번역 -> 컴파일러(Compiler)
* 유틸리티 프로그램 : 컴퓨터의 조작을 편리하게 해주는 프로그램. 기억 장치에 저장되어 필요로 할 때 호출하여 사용. (대부분은 컴퓨터 제작 회사에서 제공)

(표 3.1) 수류외 구분 
운영체제 
소프트우4어 
언어 번역 프로그램 
유틸리티 
감시 프로그램 
작업 관리 프로그램 
데이터 관리 프로그램 
문제 처리 프로그램 
MS—DOS, Windows )(P/7/8/10, UNIX, Linux 
언어 번역 프로그램 
• 어셈블러 
• 컴파일러 
• 인터프리터 
• 프리프로세서 
서비스/유털리티 
• 정렬, 조합, 편집 프로그램 
• 파일 압축 유틸리티 
• 디스크 관련 유틸리티 

+)디바이스 드라이버(책에서는 (2)시스템 소프트웨어 안에 ①시스템 소프트웨어, ②디바이스 드라이버라고.)

* 디바이스 드라이버(Device Drive) = 장치 구동기
* 하드웨어와 운영체제의 중간에 위치하여 장치를 동작시키는 일을 함.

ex) 모니터 상에 화면 출력시키는 비디오 카드 역시 출력시키기 위해선 디바이스 드라이버가 필요함.

(3) 펌웨어

**펌웨어(Firmware)** = 시스템의 효율을 높이기 위해 ROM에 들어 있는 기본적인 프로그램.

* ROM에 고정되어 있기에 하드웨어의 특성도 가지고 있으나, 실제로는 소프트웨어에 더 가깝다. 그러나 하드웨어에 장착된 칩 안에 내장된 프로그램이라는 점에서 일반적인 소프트웨어와는 다르다.
* **ex)** 바이오스(Basic Input Output System: BIOS**)** = 컴퓨터에 전원이 들어와서 초기 시작 시에 or 동작 중에 영향을 미침. 메인보드에 장착되어있다
  + 전원이 들어올 때 시스템 초기와, 이상 유무 점검, 운영체재를 적재하는(loading) 과정 담당. 운영체제 또한 동작 중에 바이오스의 일부 기능을 사용.
* 펌웨어 = 소프트웨어를 하드웨어화 시킨 것. 소프트웨어와 하드웨어의 중간에 해당.
* but 전형적인 처리 루틴, 비휘발성, 변경불가 등의 특징으로 특수한 영역에 많이 사용되고 있음.(?)

**중앙처리 장치**

중앙처리장치(CPU) = 명령을 수행하고 데이터를 처리하는 장치. 사람의 두뇌에 해당.

**입력-처리-출력**은 데이터의 기본적인 처리 과정으로서, 데이터는 처리되기 전에 입력 장치에서 주기억 장치로 읽어 들여져 기억된다. 주기억 장치에 입력된 데이터는 중앙처리 장치에 의해 처리되어 결과를 출력한다.

2.1 마이크로프로세서

마이크로프로세서 = 중앙처리 장치를 한 개의 칩으로 구현한 것.

**연산 장치** + **제어 장치** + **레지스터** + @.

우리가 통상 16비트, 32비트, 64비트 컴퓨터라고 부르는 것은 마이크로프로세서의 비트 수에 의한 것.

인텔사가 만든 **80286, 80386, 80486, 펜티엄, 펜티엄 II, 펜티엄 III, 펜티엄 IV** 등을 뜻하며, DEC사의 **알파(Alpha)**, 모토롤라사(Motorola)의 **파워피시(PowerPC)**, 썬마이크로시스템즈사의 **울트라스팍(UltraSPARC)** 등도 이에 속한다.

2.2 중앙처리 장치의 구성

중앙처리장치 = 연산장치 + 제어장치.

(1) 연산 장치

연산장치(Arithmetic and Logic Unit: ALU) = 컴퓨터의 5대 기능(입력,기억,연산,제어,출력) 중 자료를 **처리**하고 **계산**하는 장치.

수행하는 연산에는 산술연산, 논리연산이 있음.

* 산술 연산 = 덧셈, 뺄셈, 곱셈, 나눗셈 등
* 논리 연산 = 논리합(OR), 논리곱(AND), 논리부정(NOT)을 기본 연산으로 수행.

연산에 사용되는 자료는 **레지스터**라는 임시기억 장치로 가져와서 연산이 수행됨.

(2) 제어 장치

제어장치(Control Unit) = 프로그램에 의해 주어지는 연산의 순서를 차례대로 실행해나가기 위해 **기억 장치**, **연산 장치**, **입출력 장치**에 제어 신호를 보내고, 또 이들 장치로부터 신호를 받아 다음에 처리해야 할 작업들을 제어하는 역할을 한다.

제어장치의 동작)

* 첫번째 동작 : 명령어를 해독(decode)하여 제어 장치 내의 명령어 레지스터에 잠시 저장
* 두번째 동작 : 명령어 레지스터에 기억된 명령어에 따라 명령어를 실행 = 명령어 실행 단계(instruction execution cycle)

프로그램을 제어하기 위해서는 페치 단계(fetch cycle) + 해독(decode cycle)실행 단계를 거쳐야 함.

* 페치 단계 : 기억 장치의 명령어를 호출하여 명령어 레지스터에 저장
* 실행 단계 : 명령어를 해독하여 실행

2.3 중앙처리 장치의 종류

명령어의 구성 방식에 따라 **CISC**(Complex Instruction Set Computer)와 **RISC**(Reduced Instruction Set Computer) 방식으로 구분.

* CISC : 마이크로프로그래밍을 통해 다양한 명령어 형식 제공 / 구조 복잡 -> 단가 비쌈
* RISC : 연산 속도를 향상시키기 위해 제어 논리를 단순화 / 가격이 저렴, 주로 워크스테이션에 쓰임(?)

(1) CISC

* 일반적으로 사용되는 486, 586 등 인텔 계열의 중앙처리 장치의 대부분이 CISC에 속함.

명령어의 수가 많고 내부적으로 복잡한 것이 특징인데, 그에따라 발전에 한계가 있다.

(2) RISC

* 수행 속도를 높이기 위하여 처리할 수 있는 명령어의 수를 줄인 것.
* (CISC보다) 단순화된 명령어 구조, 자주 사용되는 명령어들만으로 구성 -> 수행능력의 차이는 적으나 구조의 단순화, 특별한 설계 방법을 통해 속도를 최대한으로 높일 수 있게 한 기법.

**명령어**

**페치 과정** = 주기억 장치에 저장되어 있는 프로그램 명령어(program instruction)를 호출하는 과정.

중앙처리 장치는 페치 과정을 수행하고 해독하며, 해독된 결과에 따라 자료를 이동(transfer)하고 연산과 입출력을 실행함.

* **페치 사이클(fetch cycle)** = (이 때) 명령어를 페치하는 시간.
* **실행 사이클(execution cycle)** = 명령어의 내용에 따라 수행되는 시간

(그림3.10)

**인터럽트(interrupt)** = 프로그램의 실행 도중 운영체제나 시스템에 의해서 실행 중인 프로그램이 잠시 중단된 후 특정한 작업을 수행하는 것.

* ex) 인터럽트(입출력 장치로부터 입출력의 시작이나 종료를 알려줌), 시스템 호출 인터럽트

3.1 명령어의 실행 사이클

(그림3.11)

(1) **페치 사이클**

* 프로그램 카운터가 지시하는 명령을 명령 레지스터에 페치, 명령어를 해석하여 데이터에 대한 유효 주소(effective address) 생성.

(2) **실행 사이클(execute cycle)**

* 데이터를 페치하고 명령어를 실행.

(3) **인터럽트 사이클(interrupt cycle)**

* 인터럽트가 발생하면 위의 두 사이클을 마친 후 인터럽트에 대한 처리를 함.

3.2 명령어의 실행 과정

**레지스터(register)** = 실행의 중간 결과나 적은 양의 자료를 임시로 저장하는 [그림3.12]와 같은 글립플롭(flip-flop)으로 구성된 중앙처리 장치 내의 임시기억 장치. 명령어의 실행 과정에서 매우 중요한 역할을 담당함.

(그림3.12)

레지스터 - 여러개의 비트로 이루어진 기억장치 / 회로 로서, 지정된 목적에 사용되며 수시로 그 내용물을 이용할 수 있도록 되어있음. 현재는 32비트나 64비트 레지스터를 많이 사용.

CPU 내부에는 MAR, MBR, IR, BR 등 여러 개의 레지스터가 있음. 각종의 명령을 실행할 때 명령의 내용 or 자료를 일시적으로 지하거나 하는 데 사용됨.

(그림3.14)

중앙처리 장치 내의 임시기억 장치로서의 레지스터들의 종류

1. 메모리 버퍼 레지스터(Memory Buffer Register: MBR) : 주기억 장치에 있는 명령어를 임시로 기억. MBR은 모든 자료가 거치게 되는 자료 출입용 레지스터임.
   * 명령어 = 동작을 지시하는 명령어 코드부 + 데이터가 기억된 주소를 가리키는 주소부 로 구성된 하나의 코드.
2. 메모리 어드레스 레지스터(Memory Address Register: MAR) : 주소 해독기에 의해 해독된 실행 대상의 데이터 주소(data operand)를 임시로 기억. MAR = 기억 장치 내의 정보를 호출하기 위한 주소를 기억하는 제어용 레지스터.
3. 명령어 레지스터(Instruction Register: IR) : MBR의 명령어 코드(Operation Code: OP code)를 기억. 현재 수행 중인 명령어를 임시로 기억해 두는 레지스터.
4. 프로그램 카운터(Program Counter: PC) : 다음에 실행될 명령어의 주소를 기억.
   * 주소 = 기억 장치의 특정한 위치르르 지정하기 위하여 사용되는 숫자나 이름 등의 값. 바이트나 워드 등으로 지정됨.
5. 명령어 해독기(Instruction Decoder) : 명령어 코드를 해독하여 필요한 실행 신호를 발생시킴.
   * 해독기(decoder) = 부호화된 정보를 해독하는 장치. 명령어를 해독하기 위해 쓰이는 제어장치의 일부. 그 명령어가 어떤 일을 수행하기 위한 것인가를 알아내는 장치.
6. 주소 해독기(Address Decoder) : 주소를 해독하여 그 정보를 메모리 버퍼 레지스터로 이동.
7. 인덱스 레지스터(Index Register: IR) : 명령어 실행 중 상대 주소를 결정하는 레지스터.
8. 범용 레지스터(General Purpose Register: GPR) : 사용자가 프로그램에서 다목적으로 활용할 수 있는 범용 레지스터
9. 베이스 레지스터(Base Register: BR) : 프로그램의 기준 주소를 가지고 있는 레지스터

(그림3.15)

(그림3.16)

**기억장치**

4.1 기억장치의 개념

기억장치(Memory) = 프로그램, 처리할 데이터, 처리한 결과 등을 저장하는 장치. 통상 메모리.

(1) 기억 장치의 기능

컴퓨터에서 사용하는 모든 프로그램이나 데이터를 기억시켜 두고 필요할 때에 이용할 수 있게 해줌.

**처리 속도**와 **사용속도**, **기억 용량의 크기**에 따라 **주기억 장치** / **보조기억 장치** / **레지스터** / **캐쉬(cache)** 등으로 나누어지며 계층적인 구조를 갖고 있음.(계층이 높을수록 속도up, 비트당 기억장치 비용up, 기억 용량down)

(2) 기억 장치의 분류

기억장치의 계층 구조 = 기억장치를 효율적으로 배치하여 **중앙처리 장치의 처리 속도 - I/O 속도와의 차이**를 해소하기 위한 전략.

속도가 빠를수록 비싸기에 기억장치를 적절히 배치하는 것은 시스템 전반의 성능 향상에 도움을 줌.

(그림3.17)

(3) 기억 장치의 종류와 특징

**기억 장치** = **주기억 장치** + **보조기억 장치** + **캐쉬기억 장치**. (크게 볼 때)

주기억 장치 - 주로 RAM, 보조기억 장치 - 하드디스크, CD-ROM, 테이프, 플로피 디스크, DVD, USB, SSD 등.

* **주기억 장치** : 보조기억 장치와는 달리 접근 속도가 매우 빠르다.
* **보조기억 장치** : 주기억 장치보다 용량이 크고 비용 면에서는 매우 저렴하다
* **캐쉬기억 장치** : 중앙처리 장치와 주기억 장치 간의 속도 차이를 극복하기 위한 고속의 메모리. 주기억 장치보다 고가이지만 적은 용량으로도 빠른 처리 속도를 나타냄. 주기억 장치에 있는 데이터나 내용들 중에 자주 사용되는 것을 속도가 빠른 캐쉬기억 장치에 미리 복사해 두었다가 필요할 때 중앙처리 장치에서 불러다 쓰기 때문에 매우 빠른 처리가 가능하다.

통상 가격이 비싸고 처리 속도가 매우 빠른 정적 RAM으로 구성, 기억 용량은 주기억 장치의 용량보다 매우 작음

<표 3.2>

4.2 주기억 장치

**주기억 장치** = 중앙처리 장치와의 정보 교환을 위해 프로그램과 데이터를 기억하는 장치. 중앙처리 장치와 직접 자료를 교환할 수 있는 기억 장치로서 프로그램 수행에 필요한 기본적인 명령어와 데이터를 기억.

초기의 기억소자는 자기코어(magnetic core), 지금은 초고집적도의 반도체 사용

(1) 기억 소자

* 초창기 : **진공관**을 주기억 장치의 기억 소자로 사용 - 부피가 크고, 전력 소모가 많으며, 처리 속도가 매우 느림.
* 그 후 : **자기코어**와 **트랜지스터**를 주기억 장치의 기억 소자로 사용.

자기 코어 = 아주 작은 자성물질의 링(ring), 보통 **페라이트 코어(freeite core)**를 많이 사용.

자기 코어의 한 번 자기화되면 그 자성을 계속 보존하는 성질을 이용하여 기억 장치로 사용

* 1970년대 초 : **반도체**(얇고 작은 실리콘 판에 트랜지스터를 집적)가 개발됨

현재의 주기억 장치로 이용되고 있는 반도체 기억 소자로는 RAM이 있음

* **조셉슨 소자(Josephson Junctions)** : 반도체의 실리콘 대신에 특수금속(ex.니오브(Nb))을 사용하여 극저온에서 일어나는 초전도 현상을 이용한 기억소자.

그 전의 반도체 소자보다 100배 이상 빠르며, 소비 전력도 1000배 정도가 적다.

* (여기에 분류되는건진 모르겠지만) **VRAM** : VGA 카드와 같은 비디오 회로에 사용하기 위해 설계된 칩. 텔레비전에도 사용됨.
  + 데이터를 기록하는 핀과 읽는 핀이 따로 구분되어있다 = 한 쪽에서 기록하고 있는 동안 다른 핀으로 데이터를 읽을 수 있음

(2) 주기억 장치의 구성

**주기억 장치** = **프로그램 기억 장소** + **작업 장소** + **입력 데이터 기억 장소** + **출력 데이터 기억 장소**

* **프로그램 기억 장소** = 실행할 프로그램 명령어가 기억되는 장소
* **작업 장소** = 연산 장치에서 연산된 결과를 기억하는 자옷
* **입력 데이터 장소** = 입력 장치로부터 입력된 데이터를 기억하는 장소
* **출력 데이터 기억 장소** = 출력될 데이터를 기억하는 장소

(그림3.23)

(3) 주기억 장치의 종류와 특성

집적 회로 = 작은 실리콘 판. 대량 생산 가능, 성능에 비해 저렴

주기억 장치 = 집적 회로에 기억 소자 회로를 대량 집적한 것. 크기 작음, 속도 빠름, 전력 소모 적음.

주기억 장치는 ROM과 RAM으로 나누어짐.

1. **ROM** (Read only Memory)
   * 기억된 내용을 자유롭게 읽을 수는 있지만, 데이터를 임의로 기억시킬 수 없는 읽기 전용의 비휘발성(nonvolatile memory) 기억 장치.
   * 전자 회로로 기억하기 때문에 전원이 끊어져도 내용이 보관됨 -> 지워져서는 안 될 시스템 프로그램을 기록하는데에 이용.
   * **Mask ROM** : 제조 과정에서 내용을 미리 프로그래밍하여 기억시킨 것. 사용자가 내용 변경 불가.
   * **PROM(Programmable ROM)** : 사용자가 내용을 일단 기억시킨 후에도 필요에 따라 프로그램을 다시 기억시킬 수 있는 장치. 본래 ROM은 내용이 기억되면 변경할 수 없지만, PROM은 사용자가 원하는 논리 기능을 가지도록 할 수 있다.
   * **EPROM(Erasable Programming ROM)** : PROM을 개량한 것. 자외선이나 높은 압력을 이용하여 내용 변경 가능. 전기나 자외선을 이용하여 내용을 지운 후 다시 프로그램을 기억하는 방식 -> 몇 번이고 반복 사용이 가능한 특수한 ROM이다.

(그림3.24~6)

1. **RAM** (Random Access Memory)
   * 사용자가 작성한 문장이나 프로그램이 기억되는 기억장소. 임의의 메모리 주소에 기억되어 있는 데이터를 주소 지정에 의해 즉시 판독하고 기록할 수 있는 기억장치.
   * 전원이 끊어지면 기억된 내용이 모두 지워지는 휘발성(volatile memory) 메모리. (지속적으로 유지할려면 보조기억 장치를 이용해야 함
   * 정적 RAM(SRAM), 동적 RAM(DRAM), VRAM이 있으며, 속도 빠른 순으로 나열하면 SRAM>VRAM>DRAM순.
   * 정적 RAM(Static RAM: SRAM) : 전원 공급이 되는 한 기억된 내용이 계속 남아 있는 RAM. 복잡한 재생 클럭이 필요 없어서 작은 용량의 메모리에 사용됨.
   * 동적 RAM(Dynamic RAM: DRAM) : 기억된 자료를 유지하기 위해 항상 재생용의 리프레시 펄스(refresh pulse)를 공급해주는 방식의 RAM. 정적 RAM보다 집적도, 소비 전력, 가격 면에서 우위 -> 많은 RAM을 필요로 하는 경우에 이용. 개인용 컴퓨터의 주기억 장치의 대부분은 동적 RAM.

(그림3.27? 28?~ 29 )

4.3 보조기억 장치

보조기억 장치 = 주기억 장치의 제한된 용량을 지원하는 장치.

중, 대형 컴퓨터에서는 자기 디스크(Magnetic Disk), 자기 테이프(Magnetic Tape) 등이, 개인용 컴퓨터에서는 플로피 디스크(Floppy Disk), 하드 디스크(Hard Disk), CD-ROM, DVD, SSD 등이 쓰임

(그림3.30) = 하드 디스크, CD-ROM, DVD를 차례로, <표 3.4>

(1) 중대형 컴퓨터 보조기억 장치

1. 자기 테이프 장치
   * 자기 테이프(Magnetic Tape) = 플라스틱 테이프 표면에 자성 재료인 산화철의 분말을 바른 것. 전원의 변화와 전자석의 작용에 의해 자성 분말에 자장을 만들어 반영구적인 상태로 기억시킴
2. 자기 디스크 장치
   * 자기 디스크(Magnetic Disk) - 레코드판과 같은 금속 원판을 여러 장으로 동일 축에 고정시키고, 디스크에는 원주를 따라 동심원 트랙(track)이 있고 각각의 트랙은 섹터(sector)로 나누어짐

(그림3.31)

1. 자기 드럼 장치
   * 자기 드럼