



이현진 교수님

컴퓨터개론

6주. 데이터 처리 원리

사용서체: 나눔글꼴



숭실사이버대학교

숭실사이버대학교의 강의콘텐츠는
저작권법에 의하여 보호를 받는바,
무단 전재, 복제, 배포, 전송, 대여 등을
금합니다.

지난 학습 내용



메모리, 저장장치

메모리

RAM, ROM, 플래시메모리

저장장치

자기저장방식, 광학저장방식, 고체저장방식

성능에 영향을 미치는 요인

레지스터, 메모리, 시스템클럭, 버스, 캐시



학습목차

1

데이터 표현

2

진법 변환

3

중앙처리장치

1. 데이터의 표현



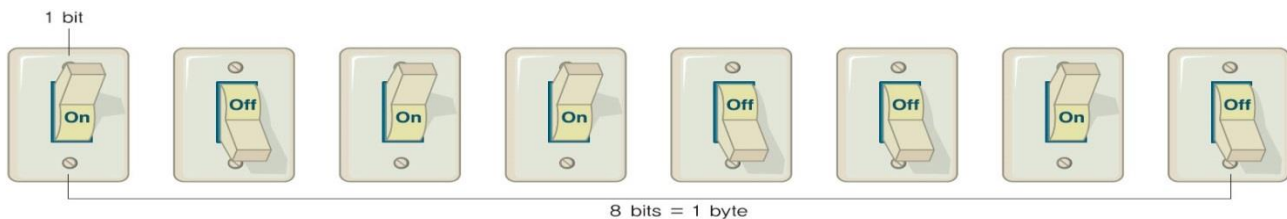
자료 표현 원리

▶ 두 가지 정보 표현

- 컴퓨터 내부에서는 전기가 흐르거나(On) 흐르지 않는(Off) 두 가지 전기 신호만을 표현할 수 있는 트랜지스터를 이용하여 자료를 처리하고 저장

▶ 2진수 체계

- 전기가 흐를 경우 '1'
- 흐르지 않을 경우 '0'



비트, 바이트

➤ 비트(bit: Binary digit)

- 컴퓨터의 정보 처리 단위 중에서 가장 작은 정보 단위

➤ 바이트(byte)

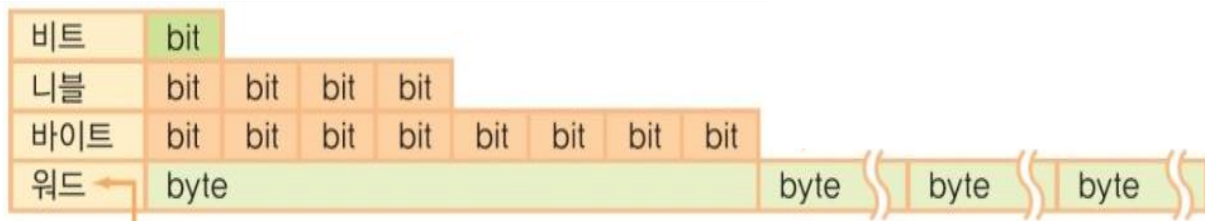
- 비트가 연속적으로 8개 모인 정보 단위

➤ 니블(nibble)

- 바이트의 1/2 크기인 4비트

➤ 워드(word)-시스템마다 크기가 다를 수 있음

- CPU가 데이터를 처리하는 단위



저장용량

▶ 바이트

- 파일이나 주기억장치, 저장장치의 크기 표현시 주로 사용
- 킬로, 메가, 기가, 테라, 페타, 엑사, 제타, 요타

표기	단위	계산	바이트 수	계량 단위
B	Byte	2^0	1	일
KB	Kilo Byte	2^{10}	1,024	천
MB	Mega Byte	2^{20}	1,048,576	백만
GB	Giga Byte	2^{30}	1,073,741,824	십억
TB	Tera Byte	2^{40}	1,099,511,627,776	조
PB	Peta Byte	2^{50}	1,125,899,906,842,624	천조
EB	Exa Byte	2^{60}	1,152,921,504,606,846,976	백경
ZB	Zetta Byte	2^{70}	1,180,591,620,717,411,303,424	십해
YB	Yotta Byte	2^{80}	1,208,925,819,614,629,174,706,176	자

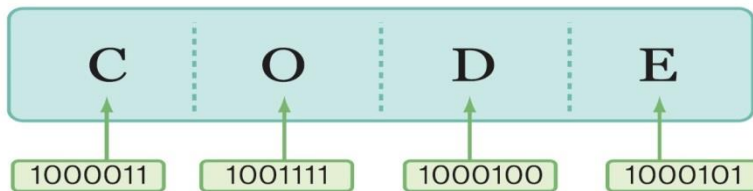
문자와 코드표

▶ 영문자는 7개의 비트의 조합으로 표현

- 일반적으로 n 비트를 사용하면 총 2^n 개의 서로 다른 문자 표현이 가능
- 각각의 조합에 일정한 문자를 할당하여 지정한 것을 문자 코드 (code)

▶ 국제 표준인 문자 코드

- 아스키코드
- 유니코드

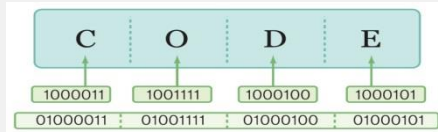


아스키코드(ASCII)

➤ 미국표준협회에서 국제적인 표준으로 정한 문자 코드 체계

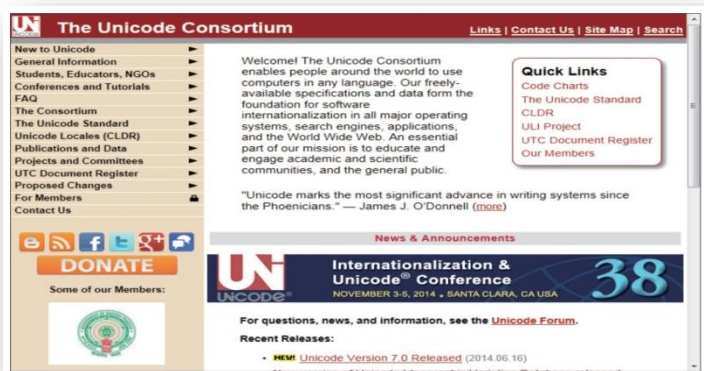
- American Standard Code for information Interchange
- 7비트로 128개의 문자, 숫자, 특수문자 코드를 규정
- 아스키코드가 7비트를 이용하지만 실제로 한 문자는 8비트인 1바이트에 저장

	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
000	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
001	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
010	Space	!	"	#	\$	%	&	'	()	*	+	,	-	.	/
011	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
100	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
101	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
110	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
111	p	q	r	s	t	u	v	w	x	y	z	{		}	~	DEL



유니코드

- ▶ 전세계 모든 언어를 하나의 코드 체계 안으로 통합하기 위하여 만들어진 코드

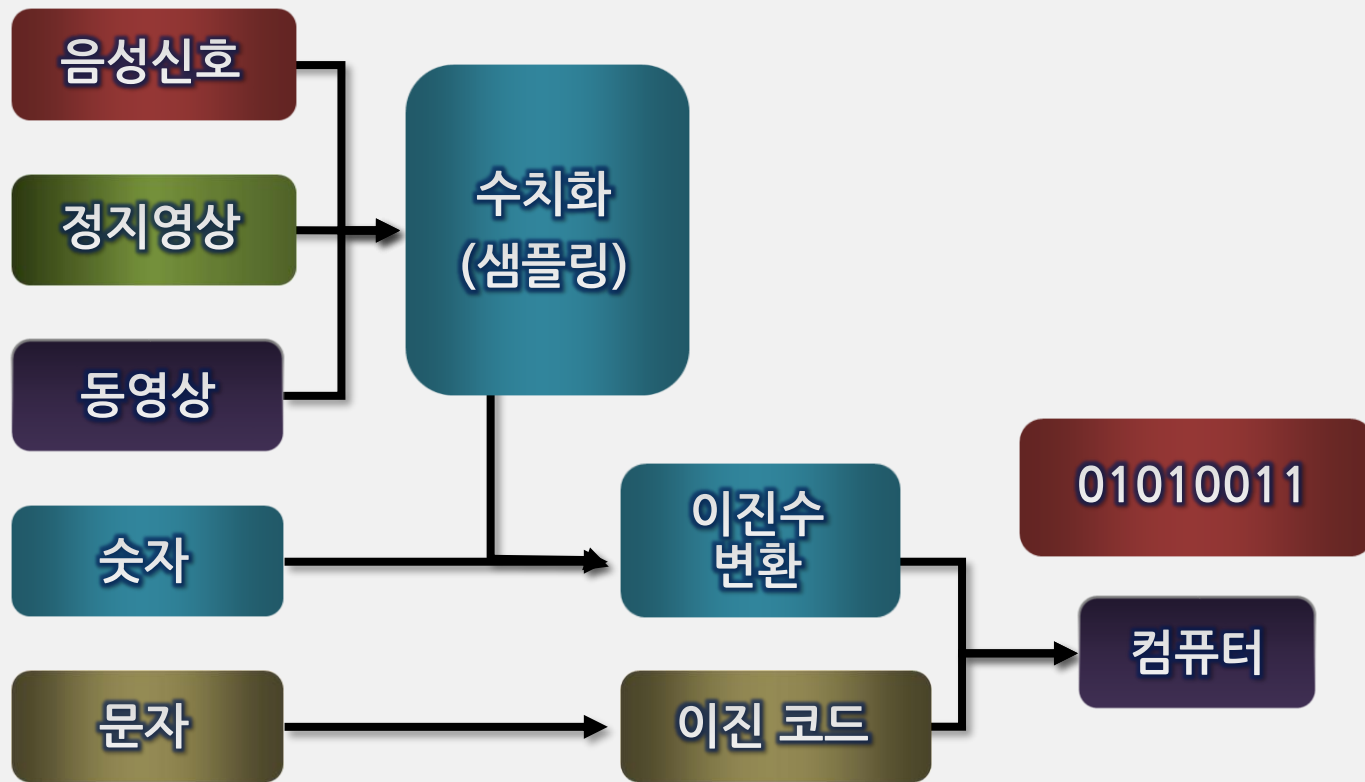


한글코드

KS 5601

- 완성형한글(2350), 한자(4888), 숫자, 특수문자 등
- 16비트 코드

컴퓨터의 데이터 표현방법



2. 진법변환



10진수

- ▶ 0에서 9까지 열 가지의 수를 한 자리(digit)의 기본 단위로 사용하는 진법
- ▶ 우리 인간이 일상 생활에서 이용하는 가장 친숙한 진수
 - 10진수의 각 자리는 오른쪽부터 1(10^0)자리, 10(10^1)자리, 100(10^2)자리 순
 - N진수인 경우 각 자릿수는 0에서 N-1까지의 정수를 이용
 - 오른쪽부터 n번째 자리의 크기는 N^{n-1}

$$\begin{array}{ccc} 2 & 5 & 6 \\ \downarrow & \downarrow & \downarrow \\ & 10^2 \text{ 자리} & 10^1 \text{ 자리} & 10^0 \text{ 자리} \end{array}$$
$$\begin{aligned} 256 &= 2 \times 10^2 + 5 \times 10^1 + 6 \times 10^0 \\ &= 200 + 50 + 6 \end{aligned}$$

2진수

➤ 0과 1의 두 가지 표현으로 각 자릿수를 표시

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1

이진수의 위치에 따른 각 자릿수의 크기

1 1 1 1 0 0 1 1

8진수, 16진수 등

➤ 8진수

- 0부터 7까지의 8가지의 수를 이용하여 숫자를 표시

$$\begin{aligned} 301_8 &= 3 \times 8^2 + 0 \times 8^1 + 1 \times 8^0 \\ &= 192 \quad + 0 \quad + 1 \\ &= 193 \end{aligned}$$

➤ 16진수

- 0에서 9까지 그리고 A에서 F까지 총 16개의 숫자나 문자를 사용하여 표시

$$\begin{aligned} 1AF_{16} &= 1 \times 16^2 + A \times 16^1 + F \times 16^0 \\ &= 256 \quad + 160 \quad + 15 \\ &= 431 \end{aligned}$$

10진수를 2진수로 변환

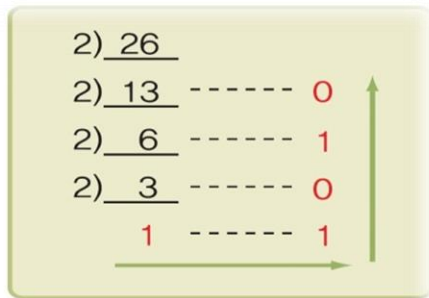
➤ 10진법->2진법 변환 단계

단계 1 : 주어진 값을 2로 나누고 그 나머지를 기록

단계 2 : 몫이 0이 아니면 계속해서 새로운 몫을 2로 나누고
그 나머지는 기록

단계 3 : 몫이 0이면 원래 값의 2진 표현은 나머지가 기록되는
순서대로 왼쪽에서 오른쪽으로 나열

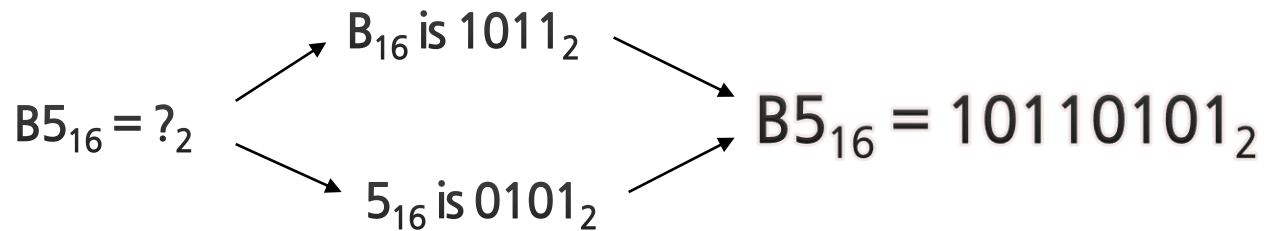
➤ $26 = 11010_2$



2)	26	
2)	13	----- 0
2)	6	----- 1
2)	3	----- 0
	1	----- 1

$26 = 11010_2$

16진수를 2진수로

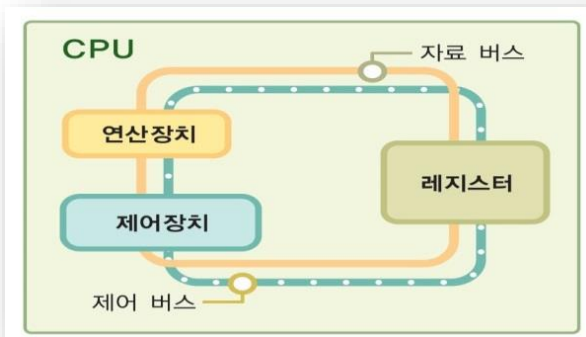


3. 중앙처리장치

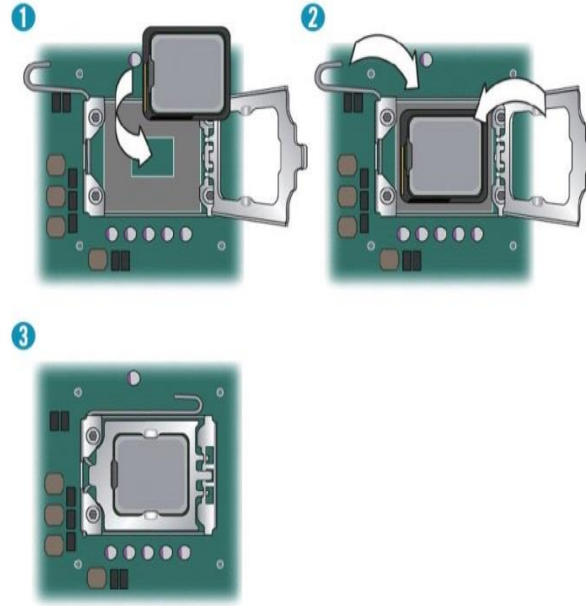
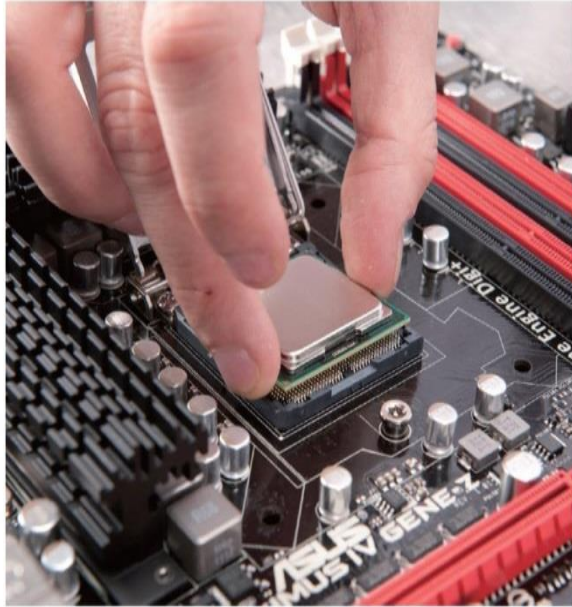


중앙처리장치 (CPU: Central Processing Unit)

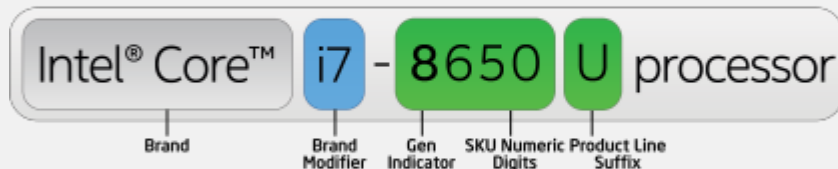
- ▶ 메모리에 저장된 프로그램과 자료를 이용하여 실제 작업을 수행하는 회로 장치
- ▶ 주요 구성 요소
 - 연산장치: 자료의 연산을 수행
 - 제어장치: 컴퓨터의 작동을 제어
 - 레지스터: 연산에 필요한 자료를 임시로 저장
 - 버스: 자료버스, 제어버스



CPU 장착하기



인텔 CPU



<https://www.intel.co.kr/content/www/kr/ko/processors/processor-numbers.html>

알파벳 접미어		설명	예
데스크탑	K	언락	제 8 세대 Intel® Core™ i7-8700 K 프로세서
모바일	G	별도 그래픽 패키지에 포함	제 8 세대 Intel® Core™ i7-8705 G 프로세서
	U	초 저전력	제 8 세대 Intel® Core™ i7-8650U 프로세서

스레드

➤ 스레드

- CPU가 독립적으로 처리하는 하나의 작업 단위
- 보통 코어 개수와 스레드 개수는 동일

➤ 하이퍼스레딩(HT) 기술

- 가상으로 운영체제가 인식하는 코어를 2배 증가시키는 방법으로 하나의 코어가 동시에 여러 스레드를 처리할 수 있는 기술

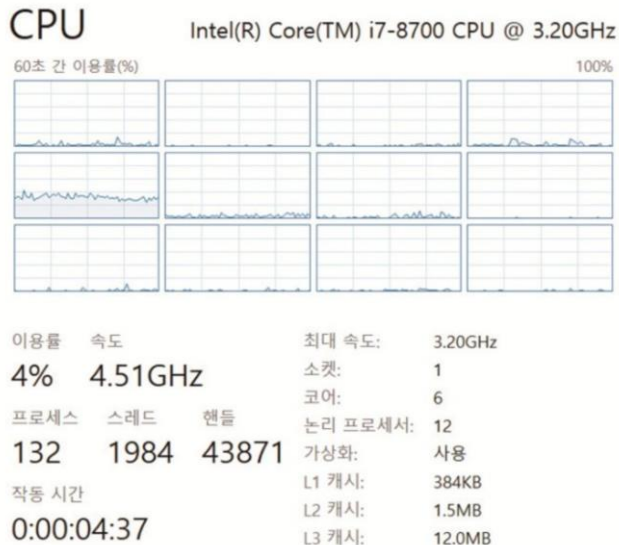
터보 부스트(Turbo Boost)

- ▶ 프로세서가 필요한 경우 동적으로 클럭 속도를 높이는 기능
- ▶ 터보 부스트로 높을 수 있는 최대 클럭 속도
 - 활성 코어의 수, 추정되는 전류 및 전력 소모량, 프로세서 온도에 따라 달라짐



본인 PC의 스레드 개수를 확인하는 방법

- Ctrl + Shift + Esc 버튼을 누르거나 작업 표시줄에서 마우스 오른쪽 버튼을 눌러 작업 관리자를 선택하면, 성능탭 'CPU 사용 현황'에서 확인 가능



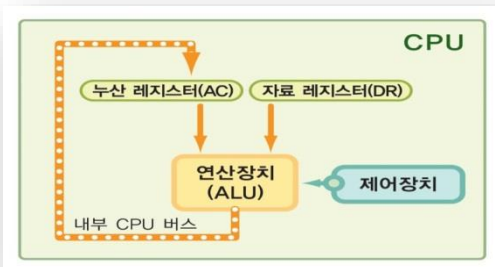
http://navercast.naver.com/magazine_contents.nhn?rid=2864&contents_id=140115

연산장치(ALU: Arithmetic and Logic Unit)

➤ 산술연산과 논리 연산을 수행하는 회로

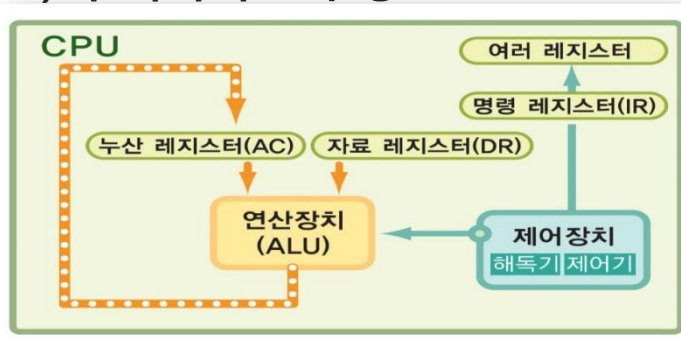
➤ 레지스터의 이용

- 누산 레지스터(Accumulator)와 자료 레지스터(Data Register)에 저장된 자료를 피연산자로 이용
- 결과는 다시 누산 레지스터에 저장되어 필요하면 주기억장치에 저장되거나 다른 연산에 이용
 - ✓ $AC \leftarrow AC + DR$
- 두 레지스터 피연산자의 연산을 연산장치가 제어장치의 신호를 받아 실행



제어장치(Control Unit)

- ▶ 신체의 여러 부분을 제어하는 인간의 뇌와 같이 제어장치는 중앙처리장치에서 연산을 제어
- ▶ 산술 및 논리 연산에 요구되는 작업을 연속적으로 수행하는 신호를 보냄으로써 연산장치와 레지스터가 명령을 수행하게 하는 장치
- ▶ 구성
 - 여러 개의 해독기(decoder)와 제어기로 구성



레지스터

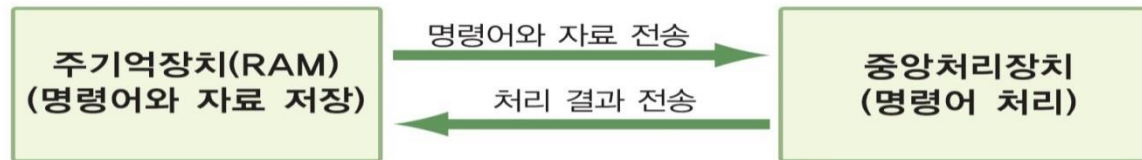
▶ 중앙처리장치 내부처리 연산에 필요한 다양한 임시 기억 장소

- 중앙처리장치 내의 레지스터 크기와 수는 중앙처리장치의 성능에 매우 중요한 요소이므로 가격과 성능을 고려하여 결정

레지스터 심볼	레지스터 이름	기능
DR	자료 레지스터 (Data Register)	연산에 필요한 피연산자를 저장하는 레지스터
AR	주소 레지스터 (Address Register)	현재 접근할 기억장소의 주소를 기억하는 레지스터
AC	누산 레지스터 (Accumulator Register)	연산장치의 입출력 데이터를 임시적으로 기억하는 레지스터
IR	명령어 레지스터 (Instruction Register)	현재 수행 중인 명령어를 저장하고 있는 레지스터
PC	프로그램 카운터 (Program Counter)	다음에 실행할 명령어의 메모리 주소가 저장된 레지스터
TR	임시 레지스터 (Temporary Register)	임시로 자료를 저장하는 레지스터로 범용 레지스터라고도 부름

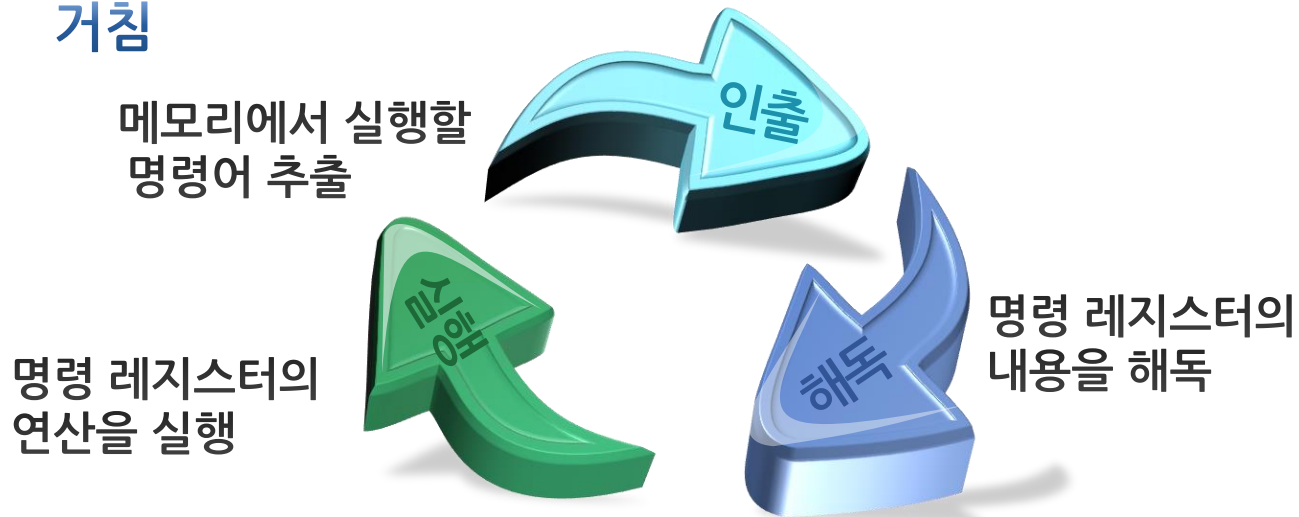
명령어 처리 과정

- ▶ 주기억장치의 명령어와 자료가 중앙처리장치의 여러 임시 저장장소인 레지스터로 전송되어 명령어를 처리한 후 다시 처리 결과인 자료가 주기억장치로 전송되는 과정을 거침



기계 주기(Machine Cycle)

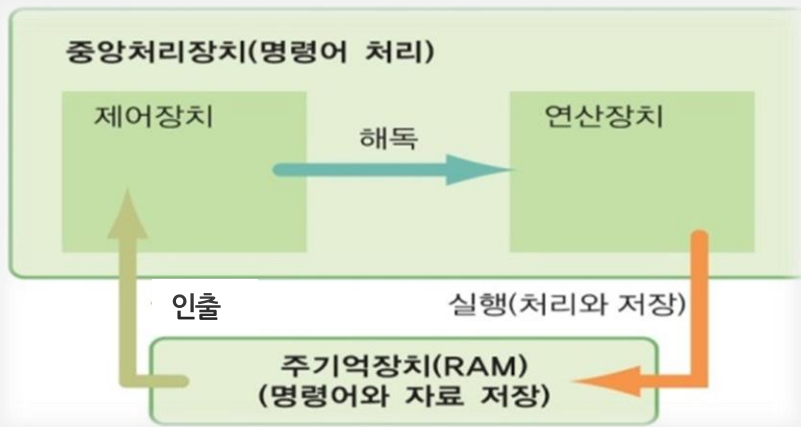
- ▶ 명령어의 집합인 프로그램을 실행하려면 세 과정인 기계 주기를 계속적으로 반복
- ▶ 중앙처리장치는 하나의 명령어를 실행하기 위하여 인출(fetch), 해독(decode), 실행(execution)의 세 과정을 거침



CPU의 처리 능력 향상

파이프라이닝

- 한 명령의 기계 사이클이 완전히 끝나기 전에 새로운 기계 사이클을 시작하는 기법





학습정리

1

데이터의 표현 비트, 바이트, 워드, 아스키코드, 유니코드

2

진법변환 10진법, 2진법, 8진법, 16진법

3

중앙처리장치 기계사이클(인출, 해독, 실행), 파이프라이닝

6주. 데이터 처리 원리



이번 강의를 마칩니다.
수고하셨습니다.