

# Introduction to Microprocessors

Jee-Hwan Ryu

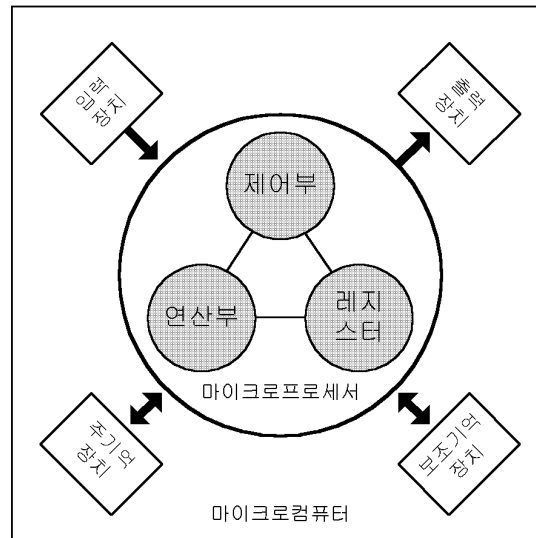
School of Mechanical Engineering  
Korea University of Technology and Education

## “마이크로프로세서” 란?

- 컴퓨터의 중앙처리장치(Central Processor Unit; CPU)를 단일 IC칩에 집적시켜 만든 반도체 소자
- 1971년 미국 Intel 사에 의해서 세계 최초로 만들어 졌으며(4004-→8008-→8080), 오늘날은 이를 흔히 MPU (MicroProcessor Unit)이라고 부르기도 한다.
- 마이크로프로세서=MPU < CPU

# 마이크로 컴퓨터

- 마이크로프로세서를 CPU로 사용하여 만든 소형 컴퓨터
- 소형경량화, 저 소비전력, 저가, 부품 수 감소로 인한 신뢰성향상



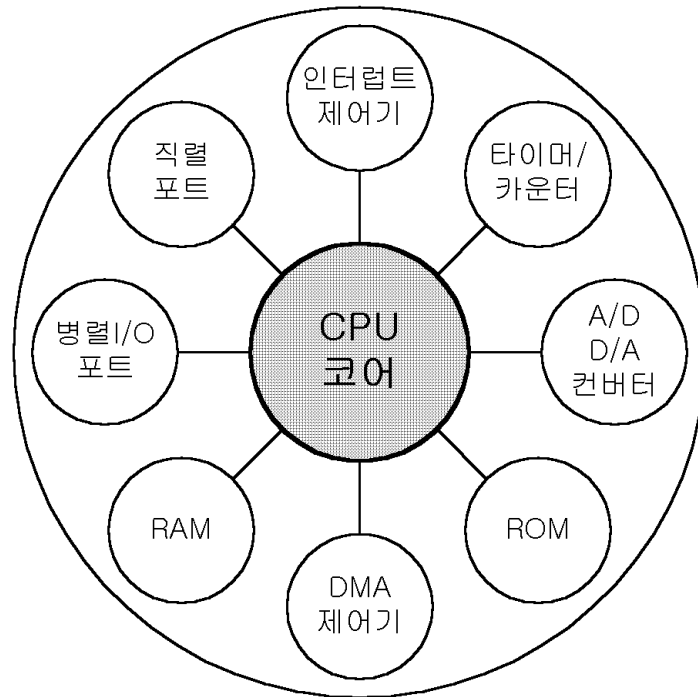
Korea University of Technology and Education

## “마이크로컨트롤러”란 ?

- 한 개의 칩 내에 CPU기능은 물론이고 일정한 용량의 메모리 (ROM, RAM 등)와 입출력 제어 인터페이스 회로까지를 내장한 것
- 범용의 목적보다는 기기 제어용에 주로 사용되므로 붙여진 이름
- Single-chip Microcomputer 라고도 부른다.
- MPU와 구분하기 위하여, MicroController Unit (MCU) 또는 MicroComputer Unit (MCU) 라고 부르기도 함
- 넓은 의미에서 마이크로프로세서는 MCU나 DSP등 CPU의 기능을 포함하는 단일 반도체 소자를 모두 지칭하는 포괄적인 개념이다.

Korea University of Technology and Education

# 마이크로컨트롤러



Korea University of Technology and Education

## 마이크로프로세서의 성능 지표

- 마이크로프로세서가 처리하는 워드(단어, word)길이
  - 4bit, 8bit, 16bit, 32bit, or 64bit
- 접근할 수 있는 메모리 크기
- 명령어 처리 속도

Korea University of Technology and Education

# Word

- 각 마이크로프로세서는 프로세서 설계를 단순화 하기 위하여 고정된 길이의 데이터를 처리
- 바이트 (1byte=8bit) 많이 사용, 워드는 마이크로 프로세서마다 크기가 다르지만, 바이트는 일정
- 4 bit 마이크로프로세서
  - 가장 저가
  - BCD 사용가능
  - 계산기, 간단한 가전기기, 장난감
- 8 bit 마이크로프로세서
  - 가장 널리 쓰임
  - 4bit의 두배, BCD 숫자 두개 다름
  - 숫자포함 문자 및 특별기호 표현 가능 (ASCII)
- 워드길이가 두 배로 되면 성능은 두 배 이상으로 강력해진다.

# Word

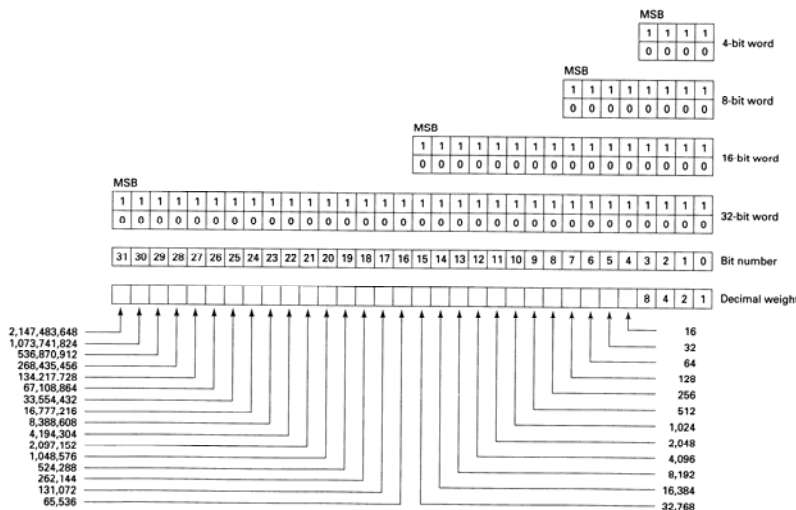


Fig. 1-5 Digital words of 4, 8, 16, and 32 bits. Note that the bit positions are numbered from right to left starting with bit 0. Bit 0 is the least significant bit, and the bit weight increases to the left. Each bit can be a binary 1 or a 0.

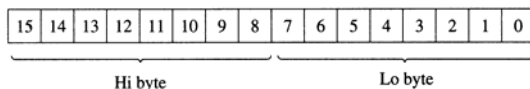


Fig. 1-6 A 16-bit digital word showing the High- and Low-byte breakdown.

## 메모리 용량

- 메모리의 데이터 워드 길이는 마이크로 프로세서에서 사용하는 워드 길이와 같다.
- 메모리 주소의 크기가 클수록 마이크로 프로세서의 성능도 커진다.
- 어떤 마이크로프로세서들은 한 개 이상의 워드를 사용하여 메모리 주소를 형성하므로, 메모리 주소범위는 마이크로 프로세서의 워드 길이에 의해서 제한되지 않는다.
- 8비트 마이크로컴퓨터에서 65,536워드 메모리 = 16비트 마이크로프로세서에서 32,768워드 메모리 = 65,536바이트

Binary address	Memory contents (4 bits long)
1 1 1 1	Data word 15
1 1 1 0	Data word 14
1 1 0 1	Data word 13
1 1 0 0	Data word 12
1 0 1 1	Data word 11
1 0 1 0	Data word 10
1 0 0 1	Data word 9
1 0 0 0	Data word 8
0 1 1 1	Data word 7
0 1 1 0	Data word 6
0 1 0 1	Data word 5
0 1 0 0	Data word 4
0 0 1 1	Data word 3
0 0 1 0	Data word 2
0 0 0 1	Data word 1
0 0 0 0	Data word 0

Fig. 1-7 A 16-word memory addressed by a 4-bit word.

## 명령어 실행 속도

- 하나의 명령어의 인출/실행 주기를 완수하는데 걸리는 시간
- 속도를 측정하는 두 가지 방법
  - 클럭 속도: 클럭의 주파수로 MHz 단위
  - 1초에 수행할 수 있는 명령어의 수: MIPS (Millions of Instructions Per Second)로 나타냄
- MIPS율은 클럭 주파수와 워드 크기에 관련
  - 동일 워드, 주파수 2배 -> 2배 MIPS
  - 워드 2배, 주파수 동일 -> 2배 MIPS
  - 평균 명령어의 크기는 1바이트 이상이고, 보통 8bit 마이크로프로세서가 명령어 1개를 인출하기 위해서는 두 번 메모리에 접근해야 함
- 벤치마크 (benchmark) 프로그램을 실행하여 비교

## 속도를 높이기 위한 기술

---

- 병렬처리
  - 두개의 프로세서 이용
- 코프로세싱
  - 전용계산을 위한 보조프로세서 이용
- 캐시 메모리 기술
  - 메모리의 접근성을 높이기 위한 중간 메모리
- 파이프라인 기법
  - 다수의 명령어들을 조립공정과 같은 방식으로 조금씩 처리

---

*Korea University of Technology and Education*

## 마이크로프로세서간의 여타 구조적 차이들

---

- 사용 가능한 레지스터 수
  - 잠시 데이터를 저장하는데 사용
- 사용 가능한 여러 종류의 레지스터들
- 여러 종류의 명령어들
- 여러 종류의 메모리 주소 지정방식
- 사용 가능한 지원 회로
  - 직렬 입출력, 카운터/타이머
- 개발 시스템 및 응용 소프트웨어와 호환성
  - 어셈블러, 고급언어, In-circuit Emulator

---

*Korea University of Technology and Education*

# 마이크로프로세서의 내부

- 블록도
  - 마이크로프로세서의 구조를 쉽게 이해하는데 필요
- 프로그램 모델
  - 마이크로프로세서의 프로그램 환경을 이해하는데 필요

## 블록도

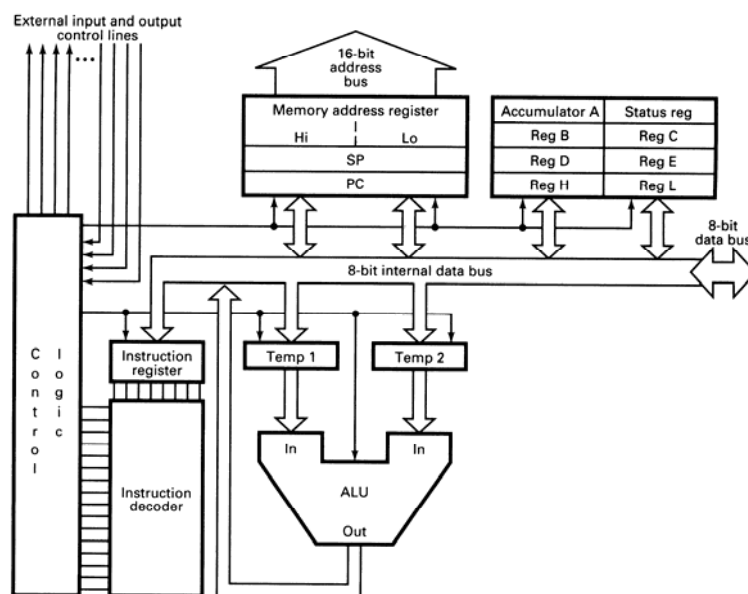


Fig. 5-1 A block diagram for the microprocessor used as a model in this chapter. This 8-bit microprocessor is used in this chapter to illustrate basic microprocessor features. These features will be found in many different forms as the more complex actual microprocessors are examined in later chapters.

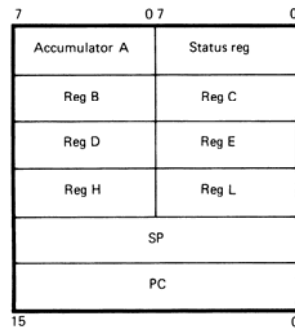


Fig. 5-2 A programming model for the 8-bit microprocessor used in this chapter. The programming model only shows those registers which the programmer can use to store data or which the programmer must access to have the right data for a program operation.

## 마이크로프로세서의 레지스터 (Register)

- 일시적으로 데이터 워드를 저장하기 위해 충분한 비트를 가진 데이터 래치
- 누산기, 프로그램 카운터, 스택 포인터, 상태 레지스터, 범용 레지스터, 메모리 어드레스 레지스터, 관련 논리, 명령 레지스터 및 임시 레지스터



# ALU (Arithmetic Logic Unit)

---

- The **arithmetic logic unit/arithmetic–logic unit (ALU)** of a computer's CPU is a part of the execution unit, a core component of all CPUs. ALUs are capable of calculating the results of a wide variety of basic arithmetical computations.
- Most ALUs will perform the following operations:
  - integer arithmetic operations
    - addition, subtraction, multiplication
  - bitwise logic operations
    - and, not, or, xor
  - bit–shifting operations
    - shifting or rotating a word by a specified number of bits to the left or right

---

*Korea University of Technology and Education*

# 누산기 (Accumulator)

---

- In a CPU, an **accumulator** is a register in which intermediate results are stored. Without a register like an accumulator, it would be necessary to write the result of each calculation (addition, multiplication, shift, etc.) to main memory, perhaps to be read right back again. Access to main memory is slower than access to the accumulator which usually has direct paths to and from the arithmetic and logic unit (ALU).

---

*Korea University of Technology and Education*

## 프로그램 카운터 (Program Counter)

---

- The **program counter** (also called the **instruction pointer** in some computers) is a register in a computer processor which indicates where the computer is in its instruction sequence. Depending on the details of the particular machine, it holds either the address of the **instruction being executed**, or the **address of the next instruction** to be executed. The program counter is automatically incremented for each instruction cycle so that instructions are normally retrieved sequentially from memory. Certain instructions, such as branches and subroutine calls, interrupt the sequence by placing a new value in the program counter.

---

*Korea University of Technology and Education*

## 상태 레지스터 (플래그 레지스터)

---

- A **status register** is a collection of flag bits for a microprocessor that indicates the status of various mathematical operations. These flags are commonly used during conditional testing and program branching.
- Different microprocessors have their own sets of status flags..
- **Common Status Register Flags**
  - **Z** = Zero flag. The result of a mathematical operation is zero
  - **C** = Carry flag. The result of an operation has produced an answer greater than the number of available bits.
  - **N** = Negative flag. The result of a mathematical operation is negative
  - **V** = Overflow flag. A register or memory location has overflowed its number of available bits. (Similar to the carry flag)
  - **I** = Interrupt flag. Interrupts can be enabled/disabled by setting or clearing this flag.

---

*Korea University of Technology and Education*

## 스택 포인터

---

- 프로그램이 돌아갈 장소를 기억하는 장소
- LIFO (last-in-first-out) 구조

## 범용 레지스터

---

- 일반적인 용도로 사용하기 위한 레지스터
- 누산기와 같은 용도로 사용되기도 함
- 쌍을 이루어 특정메모리 지정동작을 위해 이용되기도 함

# 기타 레지스터

---

- 메모리 주소 레지스터
  - 명령어를 가져올 메모리 주소를 지정한다.
- 명령 레지스터
  - 마이크로프로세서가 현재 실행하고 있는 명령어를 기억
- 임시 데이터 레지스터
  - 누산기의 저장장소를 제공