CPU and Memory

Kim, Eui-Jik

Contents

- CPU
  - CPU 정의
  - CPU 용어
  - CPU 구성요소
  - RISC CPU & CISC CPU
  - CPU 고려사항
- Memory
  - 메모리 종류
  - ROM
  - RAM
  - 프로그램 메모리와 데이터 메모리

CPU

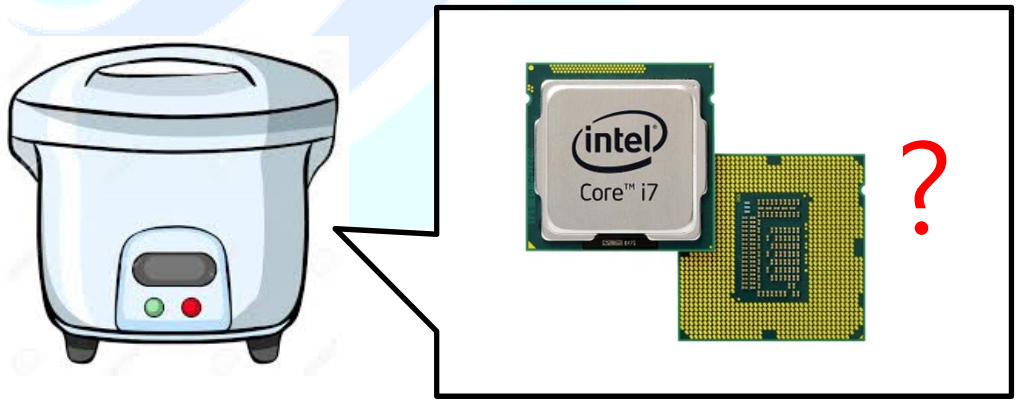
- CPU (Central Processing Unit)
  - 중앙처리장치
  - 시스템 전체를 제어하고 관리하는 역할
    - [Wikipedia] 컴퓨터 시스템을 통제하고 프로그램의 연산을 실행하고 처리하는 가장 핵심적인 컴퓨터의 제어 장치, 혹은 그 기능을 내장한 칩
  - 일반적인 CPU의 사전적 의미
    - "다양한 입력 장치 (키보드, 마우스)로부터 데이터를 받아서 처리한 후, 그 결과를 출력장치 (모니터, 프린터)로 보내는 일련의 과정을 제어하고 조정하는 장치"

Intel Core i7

AMD RYZEN 7 1800X

CPU

- 임베디드시스템의 CPU
  - "다양한 입력 방법에 의해 주어진 데이터를 받아서 처리한 후, 그 결과 데이터를 이용하여 해당 시스템의 목적에 맞는 동작이나 결과를 내도록 하는 장치"
    - 고성능의 프로세서(intel core i7 등)는 세탁기나 냉장고 등 단순한 작동만 하는 기기에서 오버 스펙임
  - 단순한 작업을 위해 만들어진 CPU
    - MPU (Micro Processing Unit)
    - MCU (Micro Controller Unit)



CPU 용어

- MPU (Micro Processing Unit)
  - 소형 처리 장치
    - 사칙연산, 논리연산, 보수 등을 주로 다루는 장치
      - 주기억장치의 비중이 큼 (큰 메모리를 요구)
    - RAM, ROM, I/O 등의 장치가 필요
    - ARM Cortex-A 시리즈 (라즈베리파이)

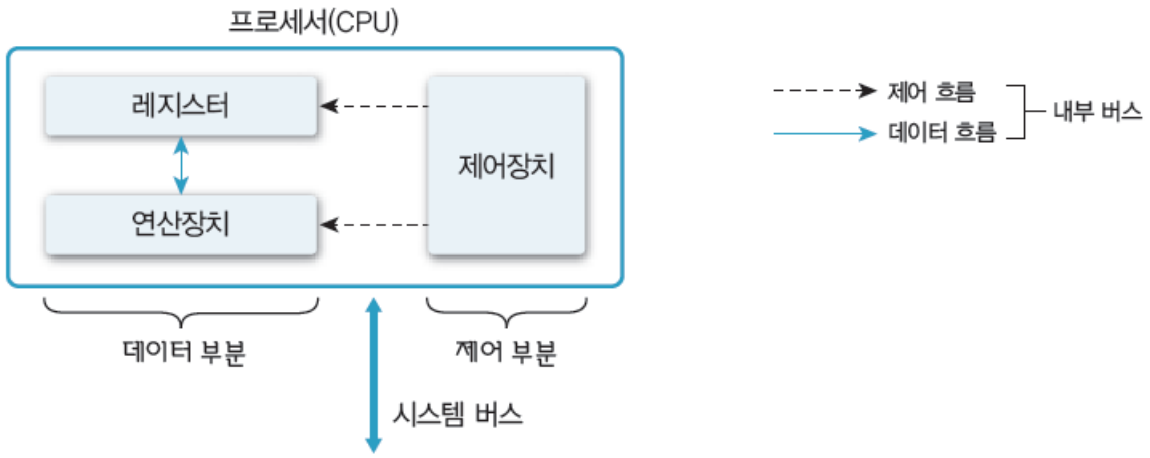
- MCU (Micro Controller Unit)
  - 소형 제어 장치
    - 집적 회로 안에 RAM, ROM, I/O, 입출력 버스 등의 최소한의 컴퓨팅 요소를 내장한 초소형 컨트롤러
    - MPU에 I/O와 메모리를 결합한 형태로 한 칩에 모든 기능을 넣음
    - ATmega 시리즈 (아두이노)

CPU 용어

- 프로세서 & 컨트롤러
  - 프로세서 (Processor)
    - 일반적으로 연산에 비중이 높은 고성능의 디바이스
    - 주로 32bit 이상의 처리 능력을 갖고 있으며, 연산에 비중이 높다는 것은 많은 양의 데이터를 처리한다는 의미
  - 컨트롤러 (Controller)
    - 일반적으로 제어에 비중이 높은 디바이스
    - 주변의 장치를 제어할 목적으로 사용하는 디바이스
    - 주로 모터 제어, 센서 제어, 간단한 디스플레이 등을 처리
    - 보통 8, 16bit의 임베디드 CPU인 경우가 많음

CPU 구성요소

- 제어 장치(CU, Control Unit) :
  - 메모리에서 명령을 받아 해독(Decode)과 실행을 지시
- 연산 장치(ALU, Arithmetic Logic Unit) :
  - 제어 장치의 지시에 따라 산술, 논리, 비트 연산 등의 실제 연산을 수행
- 레지스터 (Register) :
  - 제어, 연산 등에 사용하는 임시 기억 장치



RISC CPU & CISC CPU

- RISC (Reduced Instruction Set Computer)
  - 사용빈도가 높은 명령어만을 탑재하여 처리속도를 향상시킨 프로세서
  - 간단하고 길이가 고정된 명령어 집합을 제공
    - ARM: 32bit 고정길이 instruction set
  - S/W 복잡도 증가
  - 컴파일러 작성이 어려움
  - ex. ARM, Alpha, HP-PA, MIPS, PowerPC, Sparc

RISC CPU & CISC CPU

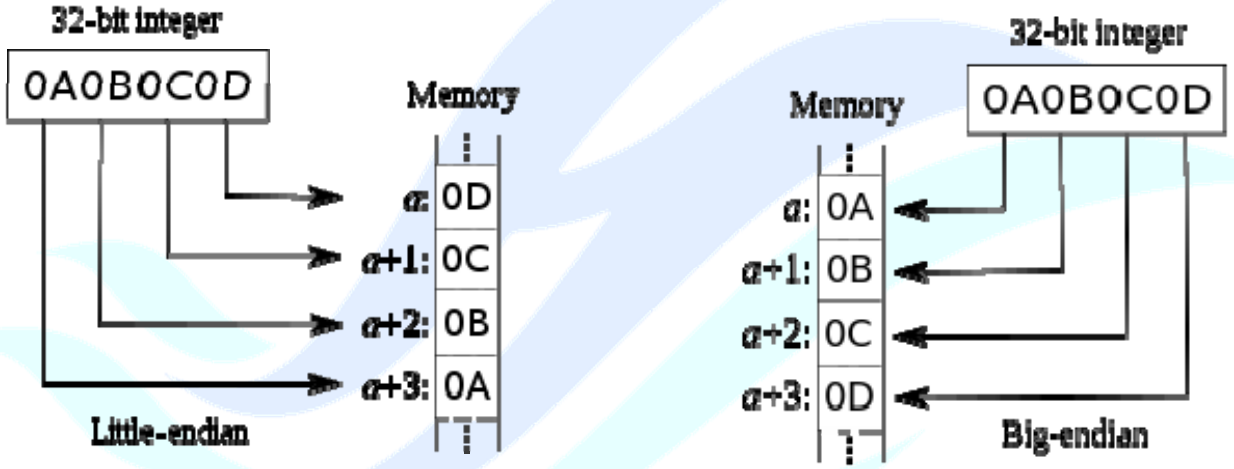
- CISC (Complex Instruction Set Computer)
  - 연산에 처리되는 복잡한 명령어들을 다수 탑재하고 있는 프로세서
  - 복잡하고 길이가 가변적인 명령어 집합을 제공
    - x86: 1~17byte 사이의 가변길이 instruction set
  - H/W 복잡도 증가
  - 컴파일러 작성이 쉬움
  - ex. x86, m68k

CPU 고려사항

- CPU 결정시 고려해야 할 사항
  - Bit number : 8-bit, 16-bit, 32-bit, 64-bit CPU
  - Clock rate : clock cycles per second (Hz)
  - Instruction set architecture : RISC or CISC
  - 엔디안
    - Endian: 메모리에 저장되는 바이트들의 순서 (Byte Order)를 나타냄
    - Little Endian: MSB (Most Significant Byte; 최상위 바이트)가 높은 주소에 위치하는 방식으로, x86 CPU가 이에 해당됨
    - Big Endian: MSB (Most Significant Byte; 최상위 바이트)가 낮은 주소에 위치하는 방식으로 RISC CPU가 이에 해당됨
  - 캐시: 내장 캐시와 외장 캐시
  - 전원: 전압/전류 규격, APM (advance power management)
  - 내장 여부: MMU (memory management unit), 레지스터, 기타 멀티미디어 처리 명령어의 내장 여부와 주변 컨트롤러를 SOC 형식으로 통합

CPU 고려사항

- 리틀엔디안(Little-endian)과 빅엔디안(Big-endian)



CPU 고려사항

- [참고] 리틀 엔디안 & 빅 엔디안
  - Little Endian
    - LSB (least significant bit)부터 저장하는 방식
    - 낮은(시작)주소에 하위 바이트부터 기록, Intel CPU 계열

예) 32비트형 4바이트 값 : 0x12345678

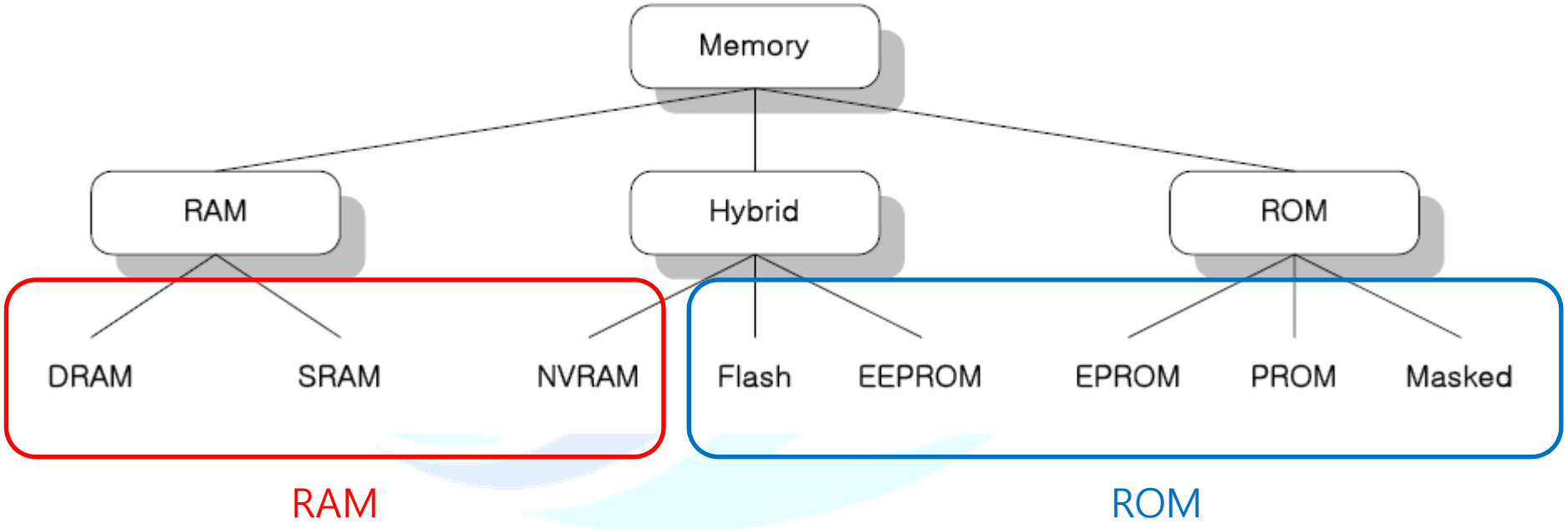
하위 주소	0x0001	0x0002	0x0003	0x0004	상위 주소
하위바이트	78	56	34	21	상위바이트
	< ----------				

  - Big Endian
    - MSB (most significant bit)부터 저장하는 방식
      - 패킷 분석을 하면서 많이 보게 되는 네트워크 프로토콜은 기본적으로 빅 엔디안 표현
    - 낮은(시작) 주소에 상위 바이트부터 기록, RISC CPU

예) 32비트형 4바이트 값 : 0x12345678

하위 주소	0x0001	0x0002	0x0003	0x0004	상위 주소
상위바이트	12	34	56	78	하위바이트
	---------- >				

메모리 종류


Memory
RAM
Hybrid
ROM
DRAM
SRAM
NVRAM
Flash
EEPROM
EPROM
PROM
Masked
RAM
ROM

메모리 종류

- ROM (Read Only Memory)
  - 읽기전용, 외부전원 없이도 내용이 유지되는 비휘발성 메모리
  - Mask ROM
    - 제품 제조/생산시 메모리에 내용을 기록하는 형태
    - 한번 기록. 즉 프로그램 한 이후에는 해당 내용에 삭제나 재 기록이 불가능
      - 제조공정에서 사용하는 마스크 패턴을 변경함으로써 정보를 기록해 넣기 때문에 유저 (user)가 직접 정보를 써넣을 수 없고 기억정보의 변경이 불가능
      - 대량 생산을 하는 경우에 경제적
  - PROM (Programmable ROM) or FPROM (Field Programmable ROM)
    - 최초 제조/생산시 메모리가 비어있고, 사용자가 단 한번 기록 가능
    - 저장공간에 퓨즈가 연결되어있으며, 데이터를 기록할 경우 퓨즈가 끊겨 데이터 수정이 불가능

메모리 종류

- EPROM (Erasable Programmable ROM) or UVEPROM (Ultra-Violet EPROM)
  - 재기록 가능. 단, 재기록 하기 위해선 메모리에 자외선을 쬐어서 내용 삭제가 가능
    - 지울때는 장치에서 메모리를 먼저 빼낸 후 지우고 나서 롬라이터 같은 장치를 이용해 다시 프로그래밍
    - 변경 횟수 10^3회 이하
- EEPROM (Electrically Erasable Programmable ROM)
  - 자외선이 아닌 전기신호를 이용해 데이터를 지우는 EPROM
    - 롬라이터 같은 별도의 장치 없이도 장착된 상태에서 내용을 지우고 기록/수정 가능
  - 기록된 내용 전체를 한번에 지우는 EPROM 과 달리 EEPROM 은 바이트 단위로 내용을 지우고 기록 가능
- Flash Memory
  - EEPROM 처럼 바이트 단위가 아닌 블록 단위 (하드디스크에서 말하는 섹터와 같은 개념)로 기록

메모리 종류

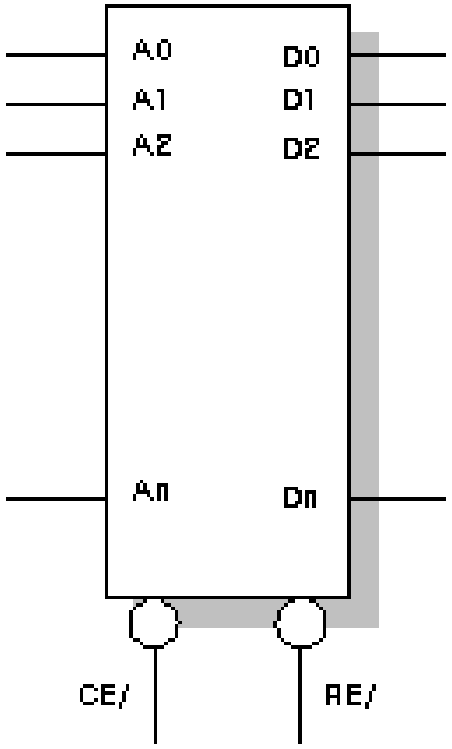
- RAM (Random Access Memory)
  - 임의접근방식에 읽기/쓰기 메모리로 휘발성 메모리
    - 기록된 데이터를 유지하기 위해선 반드시 전원이 필요
    - ROM 보다 속도가 매우 빠름
  - DRAM (Dynamic RAM)
    - 저장된 내용을 유지하기 위해 일정 간격으로 반드시 Refresh(재충전)가 필요함
    - 집적도가 높고 (전력소모 적음), 구조가 간단, 용량이 크기 때문에 일반적인 PC 용 메모리로 사용
  - SRAM (Static RAM)
    - Refresh 가 필요없고, DRAM 보다 빠름
    - 집적도가 낮고(전력소모 많음), 구조가 복잡, 용량은 작지만 속도는 아주 빠름. CPU 의 캐쉬 메모리로 주로 사용
  - NVRAM (Non-Volatile RAM)
    - 비휘발성 RAM
      - 전원이 차단되도 데이터가 사라지지 않고 유지
      - 별도의 외부 배터리가 있어서 전원이 차단되어도 데이터를 계속 유지하는 방식
      - 전원이 차단되면 EEPROM과 연동되어 해당 내용을 EEPROM 에 저장하고 전원이 다시 켜지면 해당 내용을 다시 읽어오는 방식

ROM

- ROM의 특성
  - 마이크로프로세서는 ROM으로부터 프로그램 명령어들을 읽음
  - 마이크로프로세서는 ROM에 새로운 데이터를 저장할 수 없음
    - 데이터는 변경될 수 없음
  - 전원이 꺼져도 ROM은 데이터를 기억함
  - 전원이 처음 켜질 때, 마이크로프로세서는 ROM으로부터 프로그램을 가져오기 시작함

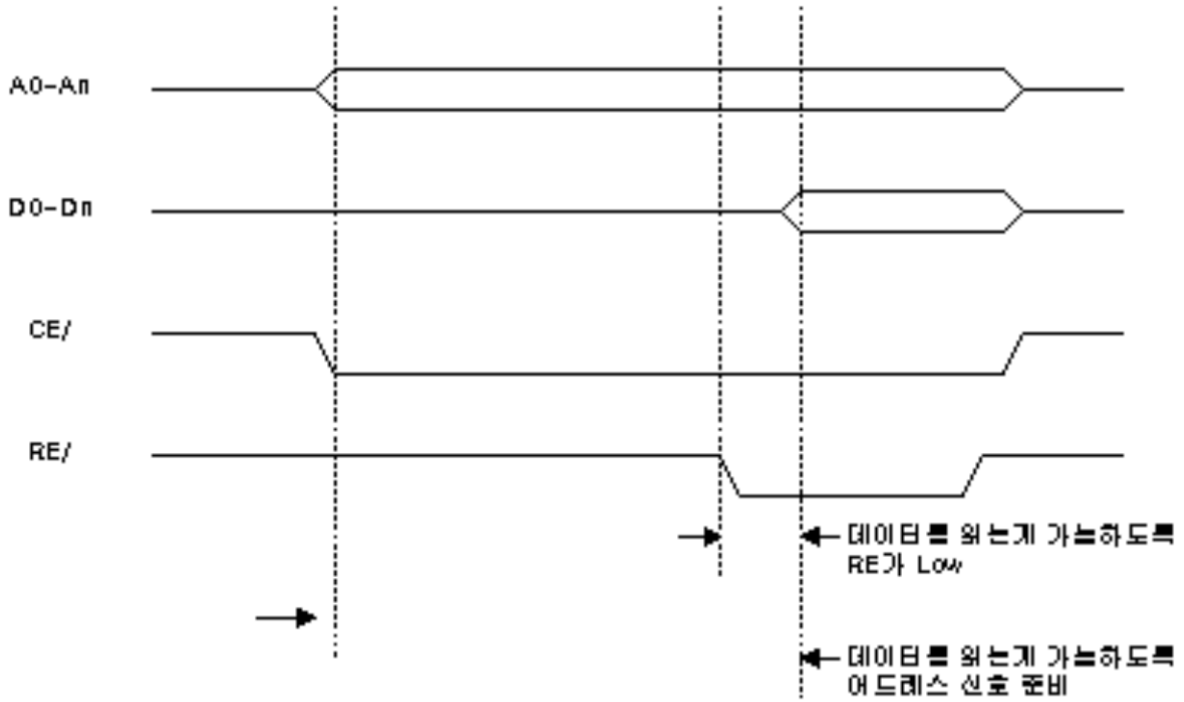
ROM

- A0 ~ An
  - 프로세서가 어디에서 데이터를 읽어올 수 있는지를 나타내는 어드레스 신호
  - 어드레스 신호선의 개수는 ROM의 크기에 따라 정해짐
- D0 ~ Dn
  - ROM으로부터 나오는 데이터 신호
  - 데이터 신호선은 일반적으로 8개 또는 16개로 구성
- CE/ 신호 (칩 인에이블 신호 or 칩 셀렉트 신호)
  - 칩 인에이블 신호로 마이크로프로세서가 ROM을 사용하길 원할 때 사용
  - 칩 인에이블 신호가 활성화되기 전까지, ROM은 어드레스 신호를 무시
- RE/ 신호
  - 리드 인에이블 신호로, ROM이 데이터들을 D0~Dn 신호선에 보내야 한다는 것을 의미
- CE/와 RE/ 신호가 활성화되지 않으면, ROM의 출력 데이터 신호선들은 하이 임피던스 상태 (플로팅된 신호선)를 유지
  - 보통 Low에서 활성화



전형적인 ROM 칩의 회로도 기호

ROM



전형적인 ROM의 타이밍 다이어그램

ROM

- 마이크로프로세서가 ROM으로부터 데이터를 읽는 단계
  - (1) 마이크로프로세서는 ROM으로부터 읽기를 원하는 데이터가 있는 주소를 어드레스 신호선에 나타냄
    - 동시에, 칩 인에이블 신호 활성화
  - (2) 리드 인에이블 신호 활성화
  - (3) ROM은 마이크로프로세서가 읽기를 원하는 데이터를 데이터 신호선에 올려놓음
  - (4) 마이크로프로세서가 데이터 신호선에서 데이터를 읽음
  - (5) 칩 인에이블 신호, 리드 인에이블 신호 비활성화

ROM

- ROM으로부터 데이터를 읽는 단계
  - 어드레스 신호선을 통해 주소를 입력하면, 해당 주소에 저장된 데이터가 데이터 신호선을 통하여 출력
  - (예제) 어드레스 신호선이 4개, 데이터 신호선이 8개인 ROM의 기억 용량?
    - 입력가능 주소는 0000, 0001, 0010, …, 1111 총 2^4개 조합
    - 각 입력된 주소를 통해 출력되는 데이터는 한 선에 1bit씩 8개의 선을 통하여 8bits (1byte) 데이터 출력
    - 따라서, ROM의 기억용량: 2^4 * 1 byte = 16 bytes

프로그램 메모리와 데이터 메모리

- 프로그램 메모리
  - CPU가 수행할 명령어를 저장해 놓은 공간, 읽기 전용 (Read-Only)
  - 일반적으로 ROM
- 데이터 메모리
  - 프로그램이 수행되는데 필요한 데이터를 저장하는 공간, 읽고 쓸 수 있음
  - 일반적으로 RAM