

AVR이란?

- 1) A(advance)V(virtual)R(RISC)로 RISC 구조의 마이크로 컨트롤러
- 2) 성능 저전력 8비트 또는 32비트 마이크로 컨트롤러
- 3) 명령어가 간단하고 동작 속도가 빠르다
- 4) 하바드 구조(Havard Architecture Architecture)에 의한 1클럭 1명령 (Assembly Assembly-language) language) 처리
 - MHz 의 클럭 주파수에서 1MIPS의 처리 속도로 동작
- 5) 다양한 명령과 쉬운 구조를 띄고 있어 마이크로컨트롤러를 이해하는데 쉽게 접근할 수 있음.
- 6) 가격이 저렴하고 응용하기 쉬워 산업시장에서도 많이 사용 되어 짐
- 7) 32개의 범용 레지스터와 RISC 구조의 디자인은 C 언어를 이용한 개발에 적합
- 8) 내부에 플래시 프로그램 메모리와 SRAM 데이터 메모리를 제공함으로써 발되는 제품의 크기를 줄일 수 있고, 제품의 크기가 줄면서 원가 절감에도 도움이 됨.
- 9) ISP(In-System Programming) 다운로드 기능 지원
- 10) JTAG 을 이용한 디버깅 용이하다

마이크로프로세서 종류

CISC

- 1) CISC는 명령어의 길이가 가변적으로 구성된 것이다.
 - 한 명령어의 길이를 줄여 디코딩 속도를 높이고 최소크기의 메모리 구조를 가집니다. 대표적으로 X86. 하드웨어의 비중이 크다.

장점

- 컴파일러 작성이 쉽다.
- 복잡한 명령도 마이크로코드(microcode)이므로 실행효율이 좋다.
- 호환성이 좋다.

단점

- 하나의 명령어가 복잡하여 해석(디코딩)에 시간이 오래 걸리며, 해석에 필요한 회로가 복잡하다.
- 적은 수의 일부 명령어만 주로 쓰인다.
- 명령어의 길이가 달라서 동시의 여러 가지의 명령처리는 어렵다.

RISC

- 1) CPU에서 수행하는 동작 대부분이 몇 개의 명령어만으로 가능하다는 사실에 기반 하여 구현함.

고정된 길이의 명령어를 사용한다.

적은수의 명령어로 명령어 집합을 구성하며 기존의 복잡한 명령은 보유한 명령어를 조합해서 사용한다.

보통 많은 수의 범용 레지스터를 가진다.

대표적으로 ARM.

소프트웨어의 비중이 크다.

장점

- 각 명령어가 한 클럭에 실행되도록 고정되어, 파이프라인 성능에 최적화
- 고정된 명령어이기 때문에 해석(디코딩)속도가 빠르며, 여러 개의 명령어를 처리할 수 있다

단점

- 컴파일러의 최적화 과정이 복잡해진다.
- 명령길이가 고정되어 있기 때문에 코드효율이 낮다.

I/O 포트를 제어하기 위해서는 각 포트당 3개의 I/O레지스터(DDRx, PORTx, PINx) 가진다.

1) DDRA(Data Direction Register)

- PORT A의 입/출력 방향설정을 위한 레지스터
- 레지스터를 SET(1)하면 출력으로 CLEAR(0)하면 입력으로 설정된다.

2) PORTA(Data Register)

- 출력용 데이터 값을 위한 레지스터

3) PINA(Input Pins Address)

- 입력 핀에 해당하는 레지스터로서 입력된 값을 표시한다.

4) SFIOR(Special Function I/O Register)

- Bit2. PUD(Pull-up Disable) : 모든 포트의 풀업저항 설정을 위한 레지스터
- 1= Disable / 0=Enable

