

인터럽트(Interrupt)

정의

프로그램을 실행하는 도중에 예기치 않은 상황이 발생할 경우 현재 실행 중인 작업을 즉시 중단하고, 발생한 상황에 대한 우선 처리가 필요함을 CPU에게 알리는 것

지금 수행 중인 일보다 더 중요한 일(ex. 입출력, 우선 순위 연산 등)이 발생하면 그 일을 먼저 처리하고 나서 하던 일을 계속해야한다.

외부/내부 인터럽트는 CPU의 하드웨어 신호에 의해 발생

소프트웨어 인터럽트는 명령어의 수행에 의해 발생

- 외부 인터럽트

입출력 장치, 타이밍 장치, 전원 등 외부적인 요인으로 발생

전원 이상, 기계 착오, 외부 신호, 입출력

- 내부 인터럽트

Trap이라고 부르며, 잘못된 명령이나 데이터를 사용할 때 발생

0으로 나누기가 발생, 오버플로우, 명령어를 잘못 사용한 경우 (Exception)

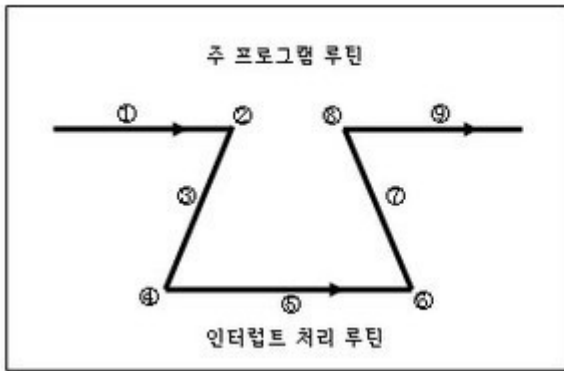
- 소프트웨어 인터럽트

프로그램 처리 중 명령의 요청에 의해 발생한 것 (SVC 인터럽트)

사용자가 프로그램을 실행시킬 때 발생

소프트웨어 이용 중에 다른 프로세스를 실행시키면 시분할 처리를 위해 자원 할당 동작이 수행된다.

인터럽트 발생 처리 과정



- ① 주 프로그램 실행
- ② 인터럽트 발생
- ③ 복귀주소 저장
- ④ 인터럽트 벡터로 점프
- ⑤ 인터럽트 처리
- ⑥ 인터럽트 처리완료
- ⑦ 복귀주소 로드
- ⑧ 마지막에 실행되던 주소로 점프
- ⑨ 주 프로그램 실행

주 프로그램이 실행되다가 인터럽트가 발생했다.

현재 수행 중인 프로그램을 멈추고, 상태 레지스터와 PC 등을 스택에 잠시 저장한 뒤에 인터럽트 서비스 루틴으로 간다. (잠시 저장하는 이유는, 인터럽트 서비스 루틴이 끝난 뒤 다시 원래 작업으로 돌아와야 하기 때문)

만약 **인터럽트 기능이 없었다면**, 컨트롤러는 특정한 어떤 일을 할 시기를 알기 위해 계속 체크를 해야 한다. (이를 ****폴링(Polling)****이라고 한다)

폴링을 하는 시간에는 원래 하던 일에 집중할 수가 없게 되어 많은 기능을 제대로 수행하지 못하는 단점이 있었다.

즉, 컨트롤러가 입력을 받아들이는 방법(우선순위 판별방법)에는 두가지가 있다.

- 폴링 방식

사용자가 명령어를 사용해 입력 핀의 값을 계속 읽어 변화를 알아내는 방식

인터럽트 요청 플래그를 차례로 비교하여 우선순위가 가장 높은 인터럽트 자원을 찾아 이에 맞는 인터럽트 서비스 루틴을 수행한다. (하드웨어에 비해 속도 느림)

- 인터럽트 방식

MCU 자체가 하드웨어적으로 변화를 체크하여 변화 시에만 일정한 동작을 하는 방식

- Daisy Chain
- 병렬 우선순위 부여

인터럽트 방식은 하드웨어로 지원을 받아야 하는 제약이 있지만, 폴링에 비해 신속하게 대응하는 것이 가능하다. 따라서 ****실시간 대응****이 필요할 때는 필수적인 기능이다.

즉, 인터럽트는 **발생시기를 예측하기 힘든 경우에 컨트롤러가 가장 빠르게 대응할 수 있는 방법**이다.