1. What is “Exception

Exception은 일반적인 프로그램 실행의 흐름을 바꾸는 몇 가지 조건을 처리할 수 있도록 설계한 프로그래밍 언어의 개념이나 하드웨어 구조를 말한다. Exception이 발생하면 순차적으로 실행되던 프로그램의 routine을 멈추고 exception handler를 실행해야한다. Exception의 종류는 reset, undefined instruction, software interrupt, prefetch abort, data abort, not assigned, IRQ, FIQ 등이 있다. Exception을 통해 프로그램 실행 도중에 발생할 수 있는 가벼운 에러를 handling해서 프로그램을 종료하지 않고도 의도한 대로 프로그램을 실행할 수 있다. Exception이 발생하면 CPSR의 값을 exception mode의 SPSR에 저장하고 PC값을 exception mode의 link register에 저장한 뒤 CPSR의 mode bit을 바꿔서 exception mode로 진입한다. 그리고 exception handler를 실행하게 된다.

1. Polling vs Interrupt

polling은 특정 주기마다 CPU가 device의 상태를 체크해 signal이 들어왔는지 확인한다. status bit value를 통해 이벤트의 발생 여부를 확인할 수 있다. Polling은 정확한 타이밍에 signal이 들어왔는지 확인할 수 없고 어느정도 오차가 있다.

Interrupt는 외부의 interrupt pin에 신호가 들어오면 그에 따른 처리를 하는 방법이다. Device가 외부로 data를 보내거나 외부로부터 data를 받으면 CPU에게 이를 직접 알리는 방식이다. Interrupt가 발생하기 전까지 CPU는 다른 작업을 수행할 수 있다. 하드웨어적으로 signal을 확인하는 방식이며 polling보다 구현이 복잡하다. 하지만 signal이 들어오는 정확한 타이밍을 확인할 수 있고 소프트웨어가 signal의 발생여부를 직접 확인할 필요가 없다.

1. Explain the process after receiving UART interrupt

ARM의 interrupt에는 normal interrupt(IRQ), fast interrupt(FIQ), software interrupt(SWI)가 있다. CPU에 여러 개의 external interrupt를 CPU의 interrupt request ports중 하나로 연결해주는 interrupt controller가 달려있다. Interrupt가 발생하면 그 interrupt에 해당하는 interrupt service routine이 실행된다. UART interrupt가 발생하면 외부로부터 전달된 data를 받게 된다. UART interrupt가 발생하면 실행 중이던 main routine을 멈추고 interrupt를 disable하고 PC값을 UART interrupt handler의 vector table entry로 설정한 뒤 CPSR의 값을 SPSR에 저장한다. 그리고 context를 저장한 뒤 UART interrupt의 interrupt service routine을 실행한다. Interrupt service routine에서 외부로부터 data를 전부 받고 나면 context를 restore하고 interrupt를 enable한 뒤 PC값과 CPSR의 값을 복원해 main routine을 다시 실행한다.