



AVR – HW2

임베디드스쿨1기

Lv1과정

2020. 08. 24

손표훈

1. ELF파일

(1) ELF파일? 실행파일, 목적파일, 공유라이브러리, 코어 덤프를 위한
유닉스 운영체계의 표준 파일 형식(Format)이다.

*윈도우에선 실행파일이나 목적파일의 파일 형식은 PE, DOS에선 COM

※ 실행파일? 코드화된(C, 어셈블리 코드와 같은..)명령에 따라 지시된 작업을 수행하도록 하는 컴퓨터 파일 즉, 게임프로그램처럼 바로 사용할 수 있는 형태

※ 목적파일? 기계어 코드로 구성되어 있는 파일로 실행파일 처럼 바로 사용할 수 없다.

※ 실행파일 vs 목적파일? 아래 그림을 참고하자!

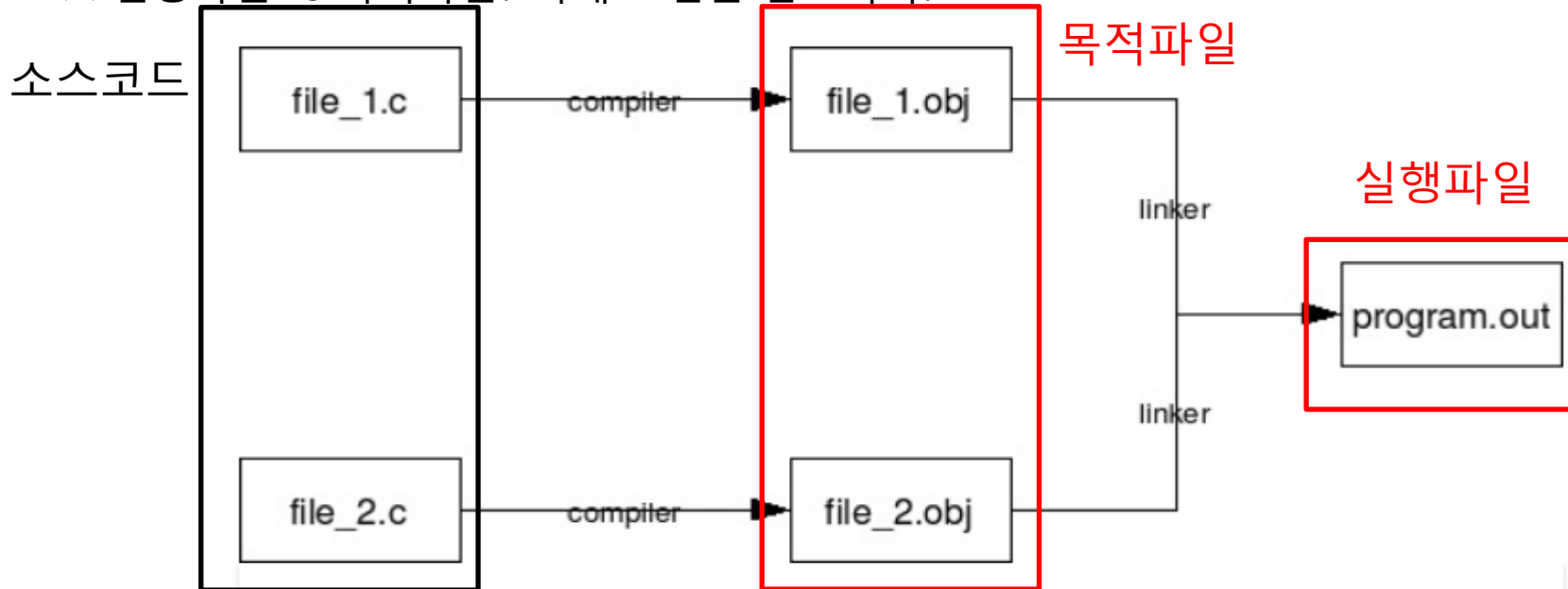
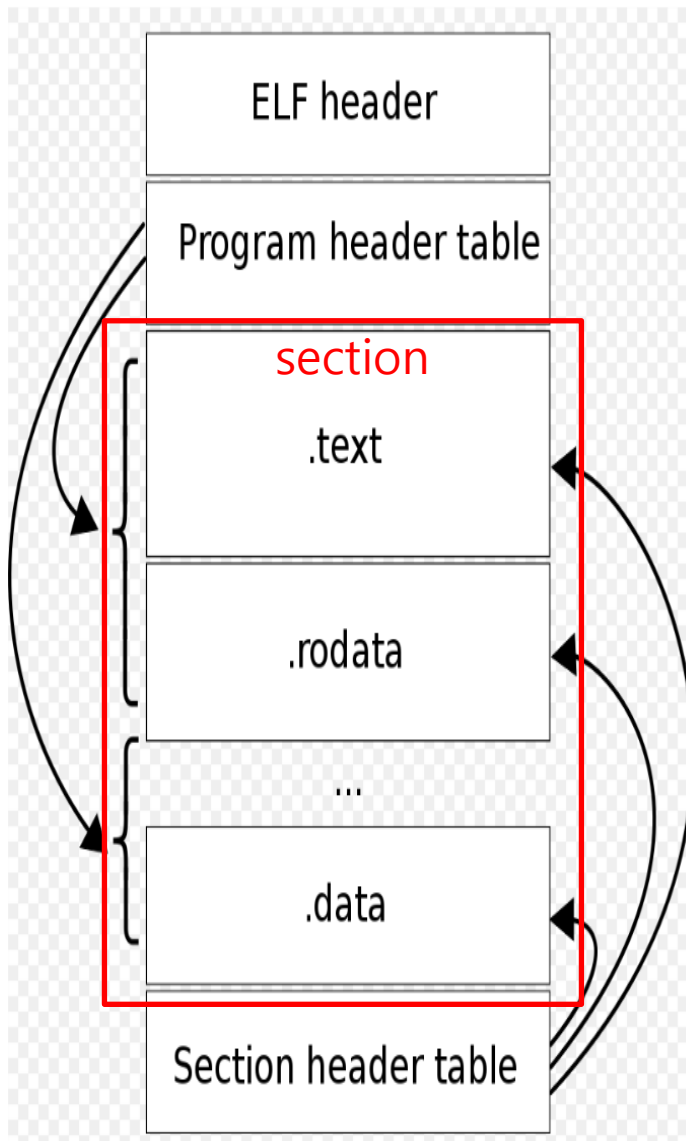


Figure 2: Creating an Executable File

1. ELF파일

(2) ELF파일구조



- ELF Header
 - > Elf 파일 포맷임을 표시하는 magic number, 이미지 형태
 - > 실행되는 CPU정보, little/big endian 정보를 포함
- Program Header table
 - > 실행 파일의 메모리 구조 내용을 표시
 - > elf 파일에서 어떤 section이 존재하고, section의 위치 정보를 포함
- .text
 - > 실제 CPU에서 수행되는 이진 기계어가 저장된 영역
- .data
 - > 프로그램 실행 시 이미 초기화 되어있는 data가 저장된 영역
- .bss
 - > 프로그램 실행 시 초기화 되어 있지 않지만, static이나 전역으로 선언된 변수가 저장된 영역
- Section영역은 C의 memory segment와 동일하게 구성된다.
- Section Header
 - > 섹션의 이름, 섹션의 크기등의 정보를 포함

2. HEX 파일

(1) HEX 파일? ELF파일과 달리 작성한 코드가 16진수 기계어로 구성되어 있으며, 실제 메모리(FLASH, EEPROM)에 다운로드 할 수 있는 파일이다.

```
/*  
 * LED_ON_OFF_CONTROL.c  
 *  
 * Created: 2020-08-29 오전 5:32:54  
 * Author : SON  
 */  
#define F_CPU 16000000L  
#include <avr/io.h>  
#include <util/delay.h>  
  
#define PORTB_REG 0x23  
  
struct io_port  
{  
    unsigned char pin; //0x23  
    uint8_t ddr; //0x24  
    uint8_t port; //0x25  
};  
  
int main(void)  
{  
    /* Replace with your application code */  
    struct io_port *portB = (void*)PORTB_REG;  
    portB->ddr = 0x20;  
  
    while (1)  
    {  
        portB->port = 0x20;  
        _delay_ms(1000);  
        portB->port = 0x00; //PORTB = 0000 0000  
        _delay_ms(1000);  
    }  
}
```



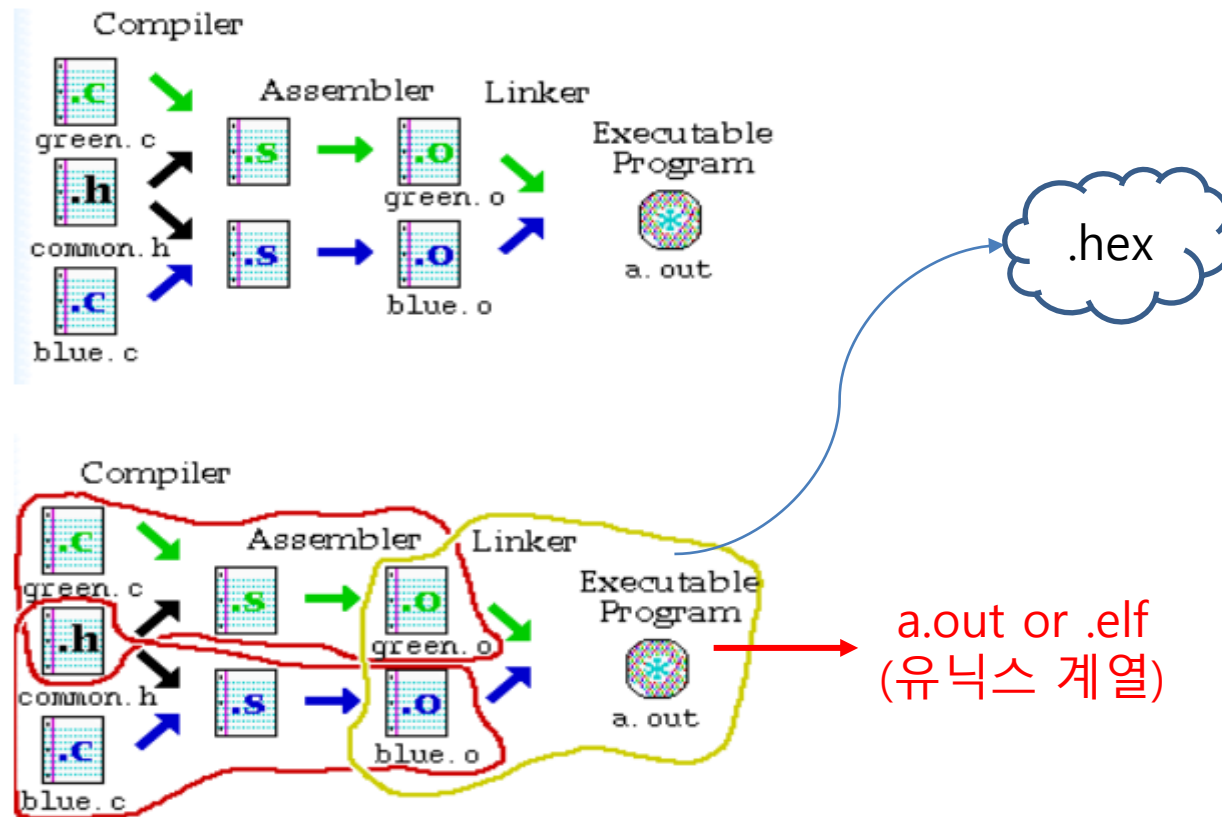
```
:10000000C9434000C943E000C943E000C943E0082  
:10001000C943E000C943E000C943E000C943E0068  
:10002000C943E000C943E000C943E000C943E0058  
:10003000C943E000C943E000C943E000C943E0048  
:10004000C943E000C943E000C943E000C943E0038  
:10005000C943E000C943E000C943E000C943E0028  
:10006000C943E000C943E0011241FBECFEFD8E04C  
:10007000DEBFCDBF0E9440000C9457000C940000DE  
:1000800080E284B985B92FEF33ED90E32150304001  
:100090009040E1F700C0000015B82FEF33ED90E37A  
:1000A000215030409040E1F700C00000EBCFF894C1  
:0200B000FFCF80  
:00000001FF
```

-> 실제 메모리 공간에 프로그램(명령어)를 업로드 하기 위한 파일형태로
.BIN(바이너리)와 목적이 같다, 다만 내용이 16진수로 변환 될 뿐...

2. HEX 파일

(2) HEX vs ELF

- ELF 파일은 HEX파일의 내용 뿐만 아니라 디버깅 정보와 지역, 전역변수의 데이터 메모리상의 위치정보를 포함하고 있다.
- HEX 파일은 ELF파일로 부터 추출된 16진수화된 기계어 코드로 구성됨

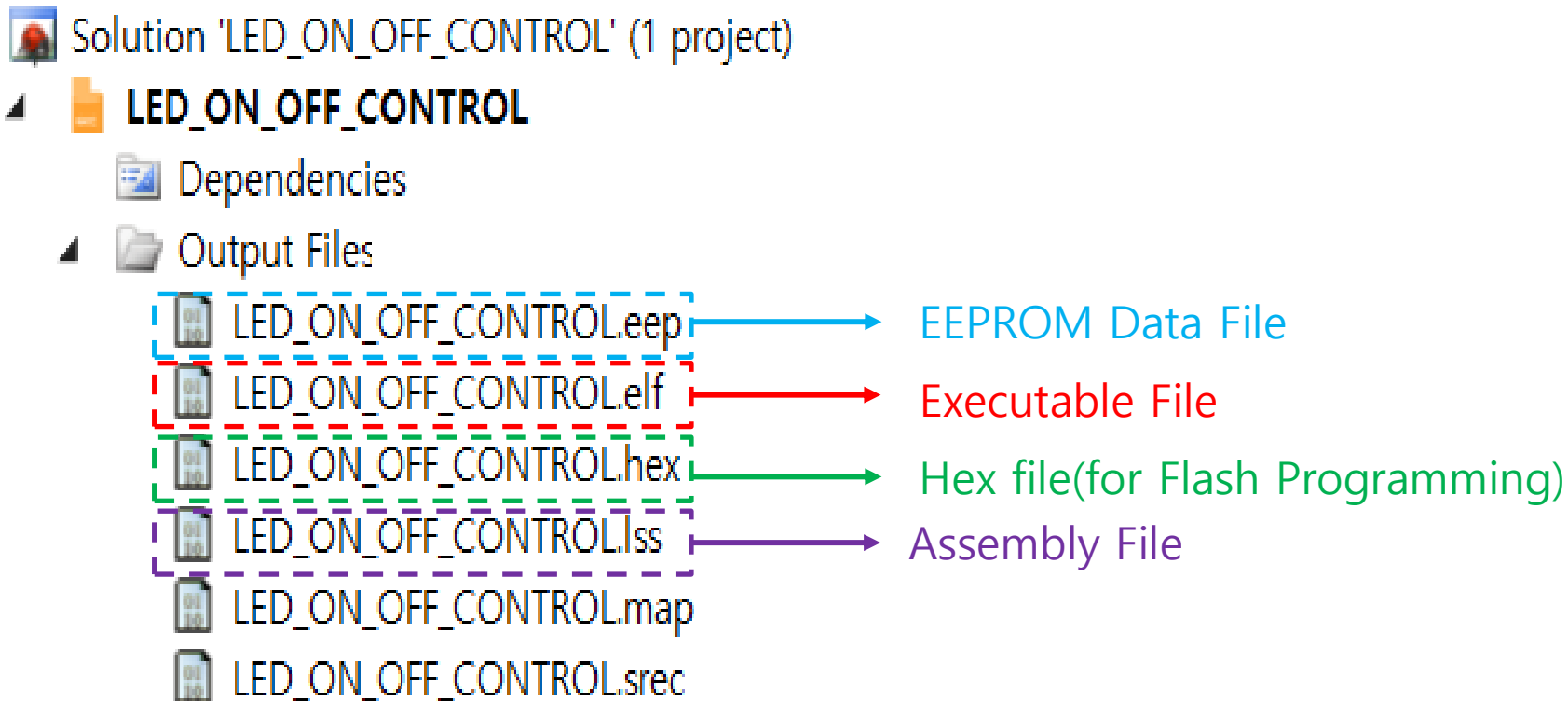


3. AVR STUDIO7 컴파일

(1) AVR STUDIO7에서 소스코드를 컴파일 하면 아래 그림과 같은 파일이 생성됨.

AVR STUDIO7는 GCC Compiler를 사용

(2) 유닉스계열의 실행파일은 a.out 또는 COFF 확장자를 사용했지만 최근 ELF로 바뀜

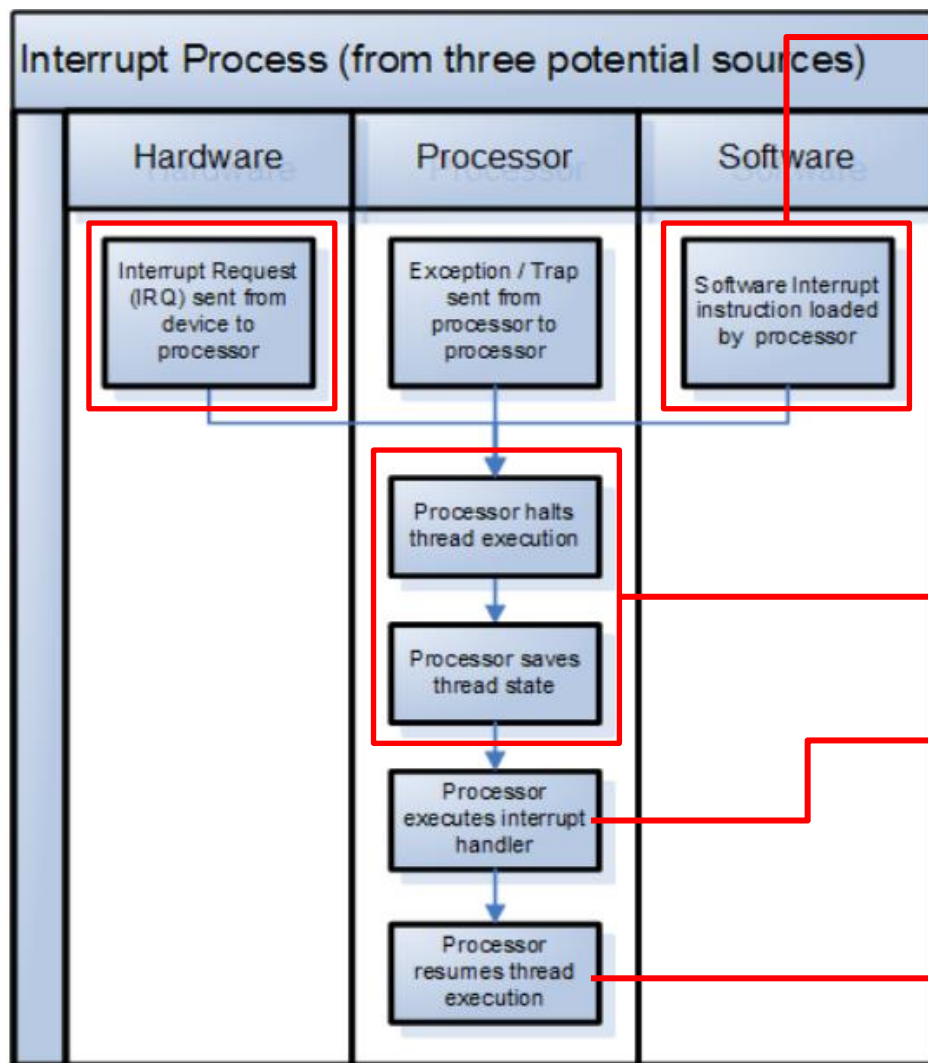


4. Interrupt

- (1) Interrupt란? CPU가 주프로그램을 실행하는 도중 주변장치(SCI, I/O, Timer/Counter 등 **Hardware Interrupt**)나 예외상황(system call, trap, program check 등 **Software Interrupt**)이 발생하여 처리가 필요한 경우 CPU에 알려 주프로그램을 중단하고 처리할 수 있도록 하는 것
- (2) 인터럽트 처리과정
 - 동시에 여러 인터럽트 발생시 처리 방법 우선순위(Daisy chain, Parallel)
 - AVR의 인터럽트 우선순위
- (3) Hardware 인터럽트
 - AVR의 Hardware 인터럽트
- (4) Software 인터럽트
 - AVR의 Software 인터럽트

4. Interrupt

(2) 인터럽트 처리과정



① 인터럽트 발생
H/W or S/W

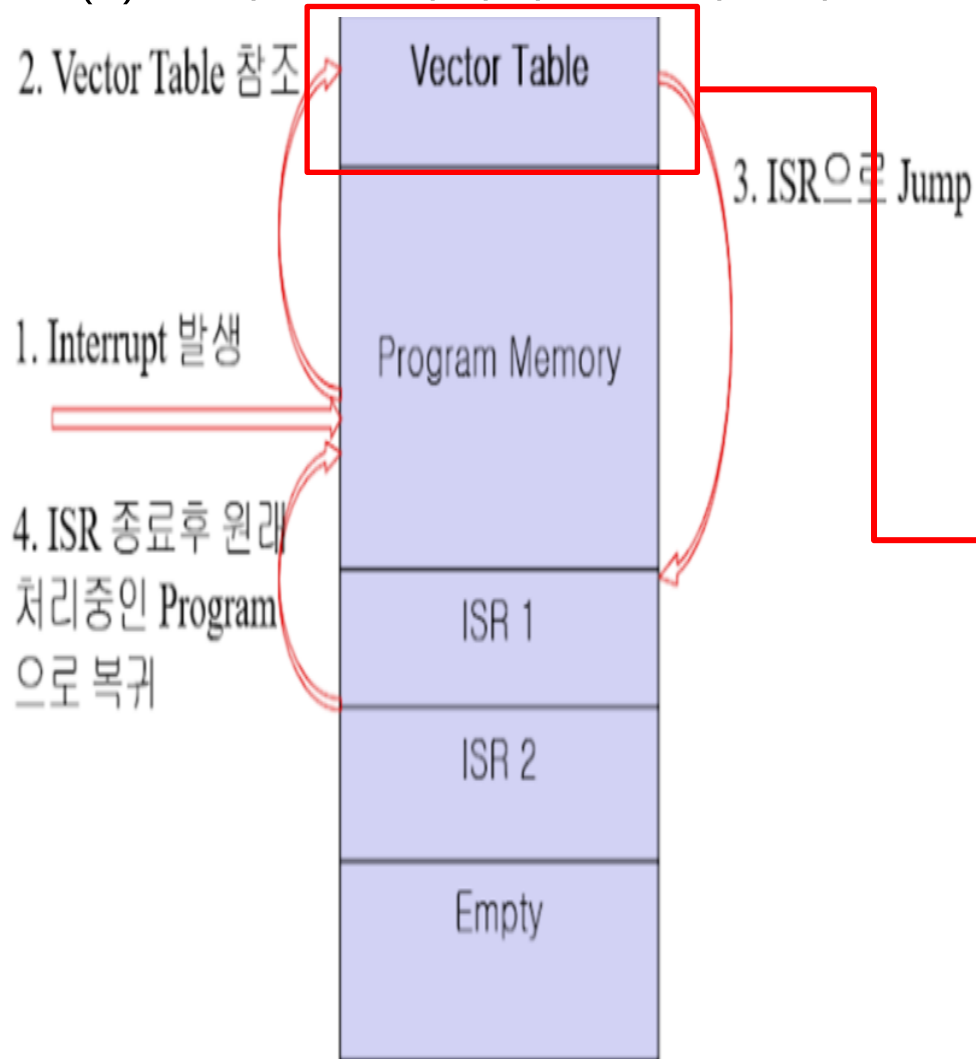
② 실행중인 명령어를 중단,
STACK에 현재 PC의 상태를 저장

③ ISR(인터럽트 서비스 루틴 = 인터럽트 핸들러)
이라는 함수를 실행

④ ISR의 명령어 실행을 마친 후 인터럽트 발생 전
처리하고 있던 명령어 실행을 재개한다.

4. Interrupt

(3) 인터럽트 처리과정을 메모리 관점에서 보면...



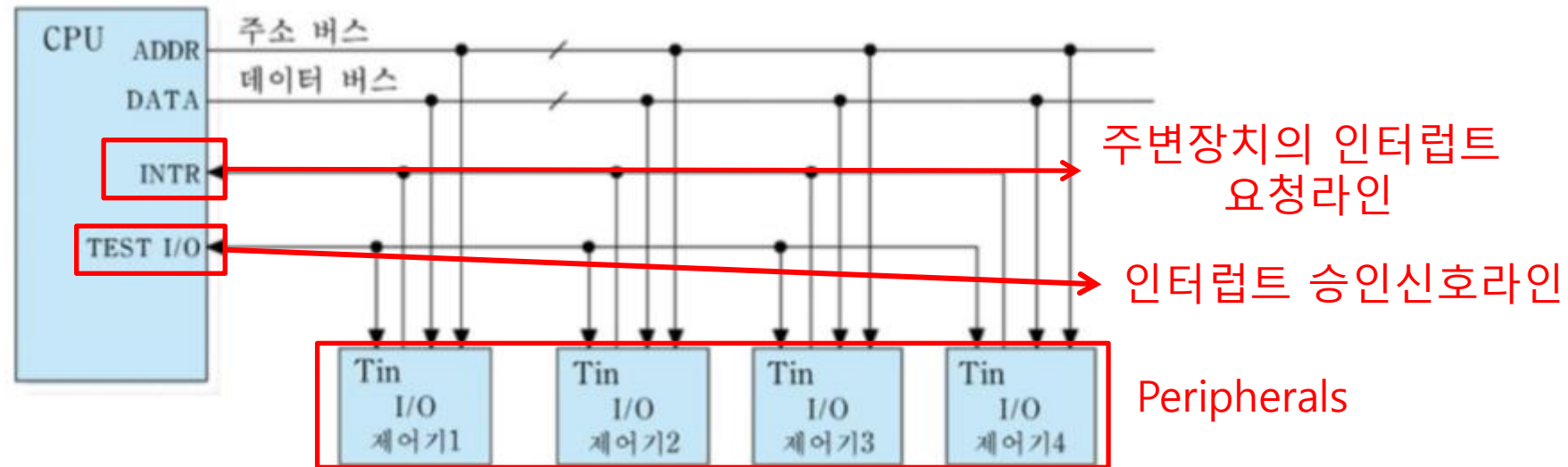
Vector No.	Program Address	Source	Interrupt Definition
1	0x0000	RESET	External pin, power-on reset, brown-out reset and watchdog system reset
2	0x0002	INT0	External interrupt request 0
3	0x0004	INT1	External interrupt request 1
4	0x0006	PCINT0	Pin change interrupt request 0
5	0x0008	PCINT1	Pin change interrupt request 1
6	0x000A	PCINT2	Pin change interrupt request 2
7	0x000C	WDT	Watchdog time-out interrupt
8	0x000E	TIMER2 COMPA	Timer/Counter2 compare match A
9	0x0010	TIMER2 COMPB	Timer/Counter2 compare match B
10	0x0012	TIMER2 OVF	Timer/Counter2 overflow
11	0x0014	TIMER1 CAPT	Timer/Counter1 capture event
12	0x0016	TIMER1 COMPA	Timer/Counter1 compare match A
13	0x0018	TIMER1 COMPB	Timer/Counter1 compare match B
14	0x001A	TIMER1 OVF	Timer/Counter1 overflow
15	0x001C	TIMER0 COMPA	Timer/Counter0 compare match A
16	0x001E	TIMER0 COMPB	Timer/Counter0 compare match B
17	0x0020	TIMER0 OVF	Timer/Counter0 overflow
18	0x0022	SPI, STC	SPI serial transfer complete
19	0x0024	USART, RX	USART Rx complete
20	0x0026	USART, UDRE	USART, data register empty
21	0x0028	USART, TX	USART, Tx complete
22	0x002A	ADC	ADC conversion complete
23	0x002C	EE READY	EEPROM ready
24	0x002E	ANALOG COMP	Analog comparator
25	0x0030	TWI	2-wire serial interface
26	0x0032	SPM READY	Store program memory ready

우선순위 ISR의 시작주소

4. Interrupt

(4) 동시에 여러 인터럽트 발생시 처리 방법 우선순위(Daisy chain, Parallel)

- Polling(SW 방식)



-> H/W가 간단하다.

-> 별도의 우선순위 점검용 Test I/O 핀을 이용하여, 인터럽트를 검사하는 방식

-> S/W적으로 인터럽트 요청 여부를 지속적으로 감시한다.

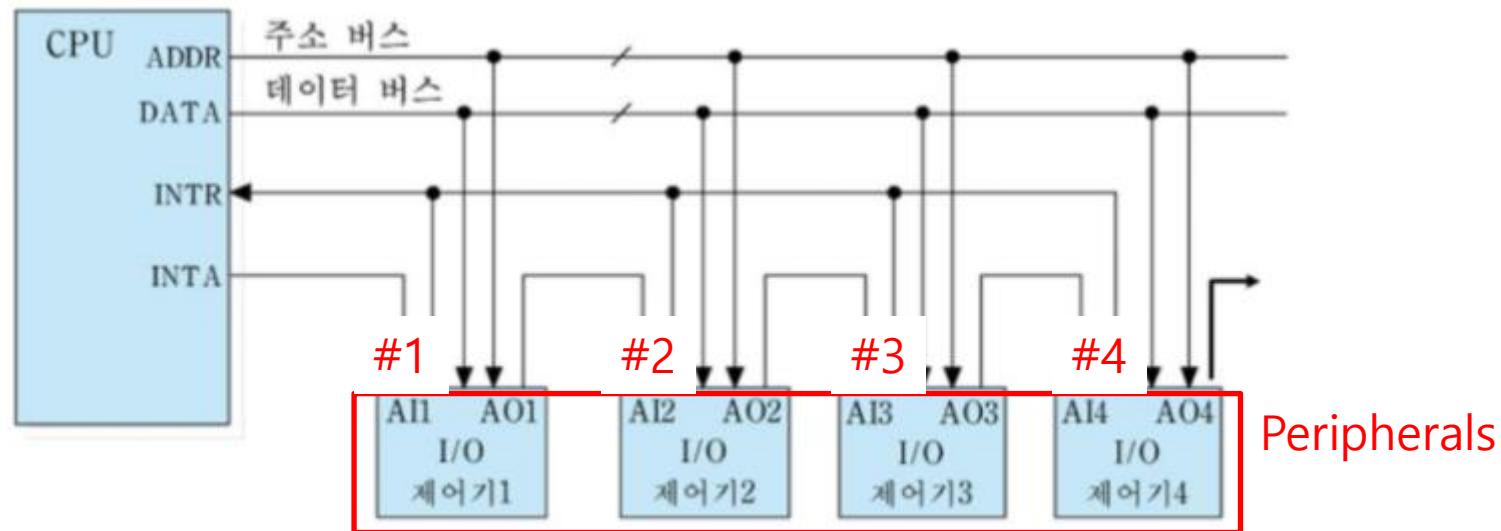
-> 우선순위 변경이 용이(검사순서를 정하면 됨)

-> 모든 장비의 인터럽트 요청을 검사하기 때문에 시간이 오래걸린다.

4. Interrupt

(4) 동시에 여러 인터럽트 발생시 처리 방법 우선순위(Daisy chain, Parallel)

- Daisy Chain방식



- > CPU의 인터럽트 활성화 신호를 Peripherals에 직렬로 연결한 형태
- > 만약 제어기1에서 인터럽트 요청이 있으면, INT A(Acknowledge)신호가 제어기 1에 인가되고, 다른 장비들에 신호를 전달하지 못하게 막는다.
- > 인터럽트 승인신호를 받으면, 제어기1은 자신의 인터럽트 ID를 데이터 버스를 통해 전달한다.
- > H/W가 간단하지만 **우선순위 낮은 장치들의 인터럽트 요청이 무시**되는 상황 발생

4. Interrupt

(4) 동시에 여러 인터럽트 발생시 처리 방법 우선순위(Daisy chain, Parallel등) - Vector Table

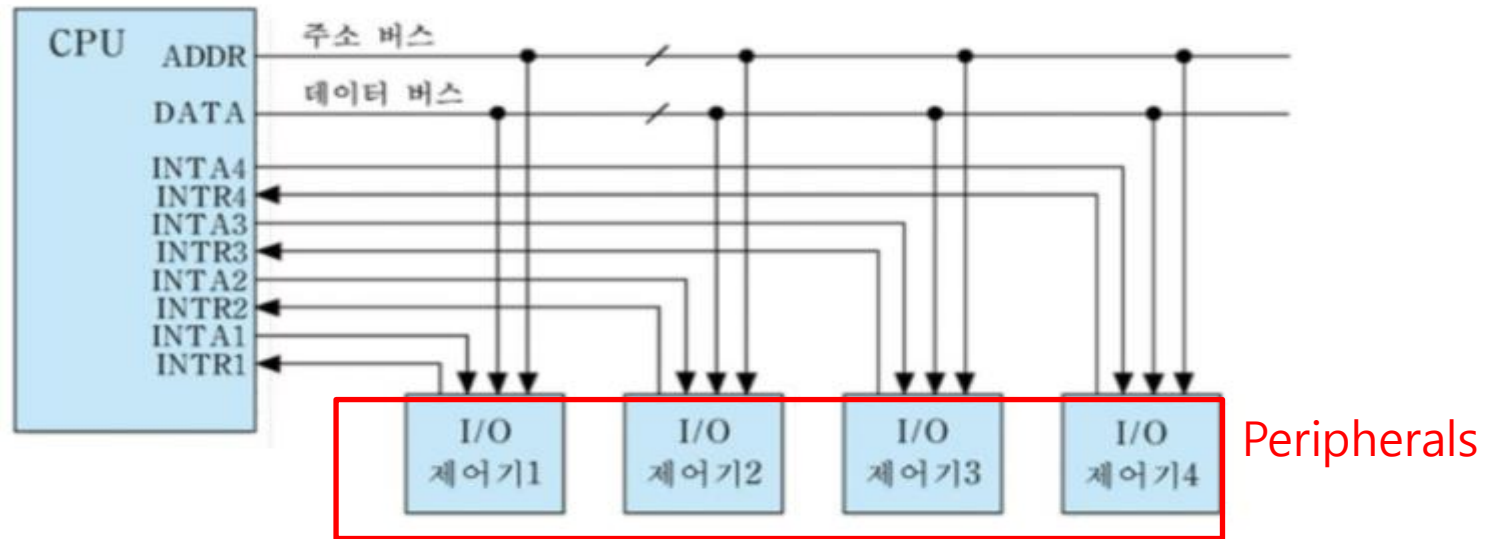
Vector No.	Program Address	Source	Interrupt Definition
1	0x0000	RESET	External pin, power-on reset, brown-out reset and watchdog system reset
2	0x0002	INT0	External interrupt request 0
3	0x0004	INT1	External interrupt request 1
4	0x0006	PCINT0	Pin change interrupt request 0
5	0x0008	PCINT1	Pin change interrupt request 1
6	0x000A	PCINT2	Pin change interrupt request 2
7	0x000C	WDT	Watchdog time-out interrupt
8	0x000E	TIMER2 COMPA	Timer/Counter2 compare match A
9	0x0010	TIMER2 COMPB	Timer/Counter2 compare match B
10	0x0012	TIMER2 OVF	Timer/Counter2 overflow
11	0x0014	TIMER1 CAPT	Timer/Counter1 capture event
12	0x0016	TIMER1 COMPA	Timer/Counter1 compare match A
13	0x0018	TIMER1 COMPB	Timer/Counter1 compare match B
14	0x001A	TIMER1 OVF	Timer/Counter1 overflow
15	0x001C	TIMER0 COMPA	Timer/Counter0 compare match A
16	0x001E	TIMER0 COMPB	Timer/Counter0 compare match B
17	0x0020	TIMER0 OVF	Timer/Counter0 overflow
18	0x0022	SPI, STC	SPI serial transfer complete
19	0x0024	USART, RX	USART Rx complete
20	0x0026	USART, UDRE	USART, data register empty
21	0x0028	USART, TX	USART, Tx complete
22	0x002A	ADC	ADC conversion complete
23	0x002C	EE READY	EEPROM ready
24	0x002E	ANALOG COMP	Analog comparator
25	0x0030	TWI	2-wire serial interface
26	0x0032	SPM READY	Store program memory ready

-> 인터럽트의 우선순위와 인터럽트 핸들러(ISR)의 주소를 그림과 같이 미리 정의해 놓는 방식

-> 인터럽트 요청 발생시 우선순위에 따라 각 핸들러의 주소로 점프하여 인터럽트를 처리

4. Interrupt

- (4) 동시에 여러 인터럽트 발생시 처리 방법 우선순위(Daisy chain, Parallel등)
- Parallel 방식



- > 각 장치가 개별 인터럽트 라인을 가진다.
- > 우선순위는 **Mask Register의 비트 위치**에 의해서 결정된다.
- > Mask Register는 우선순위가 높은 인터럽트를 실행 중에 우선순위가 낮은 인터럽트를 비활성화 시킬 수 있다.
- > 우선순위가 낮은 인터럽트 실행 중 우선순위가 높은 인터럽트가 실행된다.

4. Interrupt

(5) Atmega328P의 인터럽트

Vector No.	Program Address	Source	Interrupt Definition
1	0x0000	RESET	External pin, power-on reset, brown-out reset and watchdog system reset
2	0x0002	INT0	External interrupt request 0
3	0x0004	INT1	External interrupt request 1
4	0x0006	PCINT0	Pin change interrupt request 0
5	0x0008	PCINT1	Pin change interrupt request 1
6	0x000A	PCINT2	Pin change interrupt request 2
7	0x000C	WDT	Watchdog time-out interrupt
8	0x000E	TIMER2 COMPA	Timer/Counter2 compare match A
9	0x0010	TIMER2 COMPB	Timer/Counter2 compare match B
10	0x0012	TIMER2 OVF	Timer/Counter2 overflow
11	0x0014	TIMER1 CAPT	Timer/Counter1 capture event
12	0x0016	TIMER1 COMPA	Timer/Counter1 compare match A
13	0x0018	TIMER1 COMPB	Timer/Counter1 compare match B
14	0x001A	TIMER1 OVF	Timer/Counter1 overflow
15	0x001C	TIMER0 COMPA	Timer/Counter0 compare match A
16	0x001E	TIMER0 COMPB	Timer/Counter0 compare match B
17	0x0020	TIMER0 OVF	Timer/Counter0 overflow
18	0x0022	SPI, STC	SPI serial transfer complete
19	0x0024	USART, RX	USART Rx complete
20	0x0026	USART, UDRE	USART, data register empty
21	0x0028	USART, TX	USART, Tx complete
22	0x002A	ADC	ADC conversion complete
23	0x002C	EE READY	EEPROM ready
24	0x002E	ANALOG COMP	Analog comparator
25	0x0030	TWI	2-wire serial interface
26	0x0032	SPM READY	Store program memory ready

GPIO 인터럽트

Watchdog 인터럽트

Timer/Counter 인터럽트

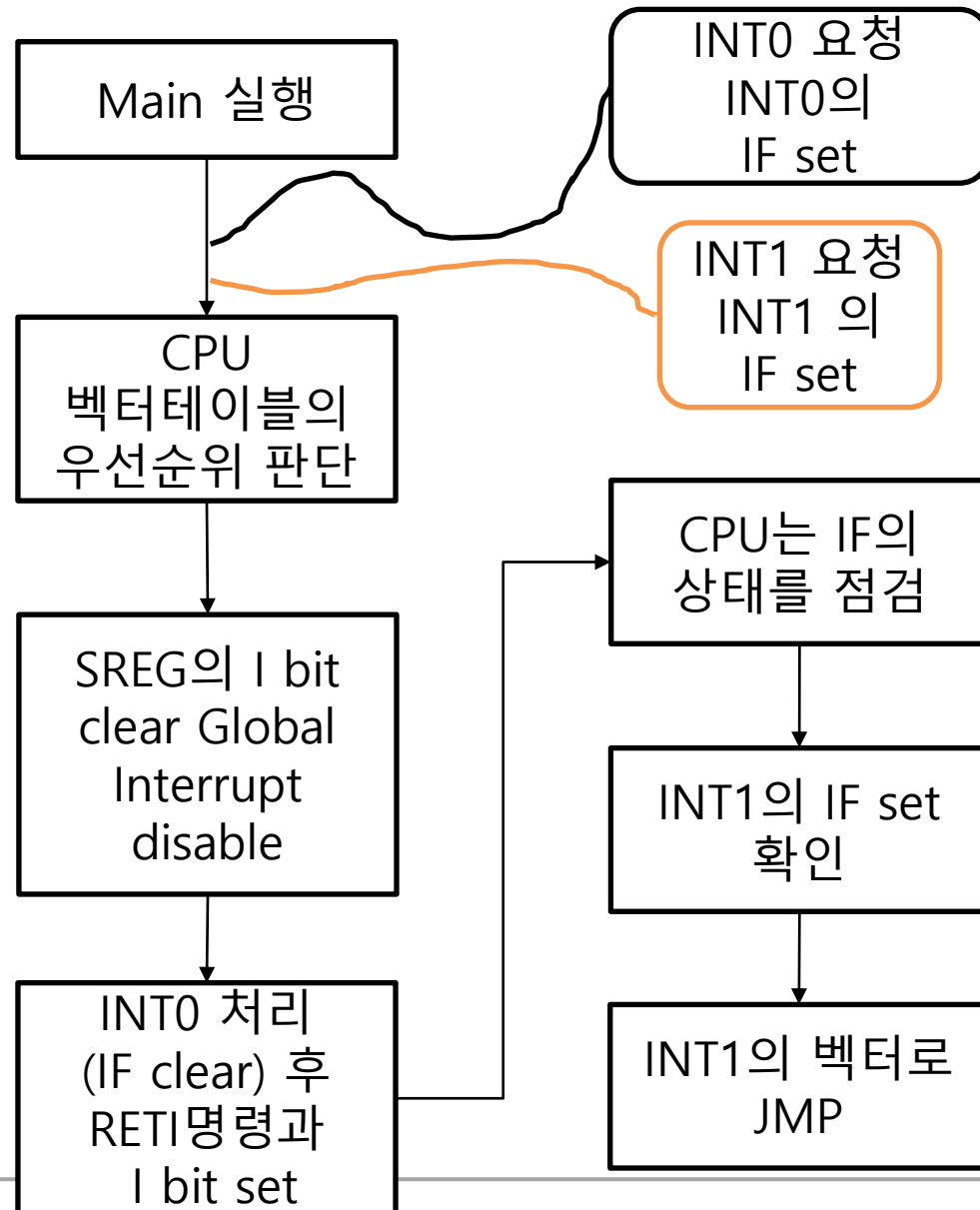
ADC 인터럽트

통신관련 인터럽트

우선순위 ISR의 시작주소

4. Interrupt

(6) Atmega328P에서 동시에 여러 인터럽트가 발생하면? *IF = Interrupt Flag Register



Vector No.	Program Address	Source	Interrupt Definition
1	0x0000	RESET	External pin, power-on reset, brown-out reset and watchdog system reset
2	0x0002	INT0	External interrupt request 0
3	0x0004	INT1	External interrupt request 1
4	0x0006	PCINT0	Pin change interrupt request 0
5	0x0008	PCINT1	Pin change interrupt request 1
6	0x000A	PCINT2	Pin change interrupt request 2
7	0x000C	WDT	Watchdog time-out interrupt
8	0x000E	TIMER2 COMPA	Timer/Counter2 compare match A
9	0x0010	TIMER2 COMPB	Timer/Counter2 compare match B
10	0x0012	TIMER2 OVF	Timer/Counter2 overflow
11	0x0014	TIMER1 CAPT	Timer/Counter1 capture event
12	0x0016	TIMER1 COMPA	Timer/Counter1 compare match A
13	0x0018	TIMER1 COMPB	Timer/Counter1 compare match B
14	0x001A	TIMER1 OVF	Timer/Counter1 overflow
15	0x001C	TIMER0 COMPA	Timer/Counter0 compare match A
16	0x001E	TIMER0 COMPB	Timer/Counter0 compare match B
17	0x0020	TIMER0 OVF	Timer/Counter0 overflow
18	0x0022	SPI, STC	SPI serial transfer complete
19	0x0024	USART, RX	USART Rx complete
20	0x0026	USART, UDRE	USART, data register empty
21	0x0028	USART, TX	USART, Tx complete
22	0x002A	ADC	ADC conversion complete
23	0x002C	EE READY	EEPROM ready
24	0x002E	ANALOG COMP	Analog comparator
25	0x0030	TWI	2-wire serial interface
26	0x0032	SPM READY	Store program memory ready

Branch Instructions					
RJMP	k	Relative jump	$PC \leftarrow PC + k + 1$	None	2
IJMP		Indirect jump to (Z)	$PC \leftarrow Z$	None	2
JMP	k	Direct jump	$PC \leftarrow k$	None	3
RCALL	k	Relative subroutine call	$PC \leftarrow PC + k + 1$	None	3
ICALL		Indirect call to (Z)	$PC \leftarrow Z$	None	3
CALL	k	Direct subroutine call	$PC \leftarrow k$	None	4
RET		Subroutine return	$PC \leftarrow STACK$	None	4
RETI		Interrupt return	$PC \leftarrow STACK$	I	4
CPSE	Rd, Rr	Compare, skip if equal	if (Rd = Rr) $PC \leftarrow PC + 2$ or 3	None	1/2/3
CP	Rd, Rr	Compare	$Rd - Rr$	Z, N, V, C, H	1

4. Interrupt

(7) H/W 인터럽트?

- CPU 외부의 주변장치(Peripherals)에 의해 발생하는 인터럽트
- 비동기식 인터럽트이다 : 주변장치들의 상태에 의해 예상치 못하게 발생하기 때문

Vector No.	Program Address	Source	Interrupt Definition
1	0x0000	RESET	External pin, power-on reset, brown-out reset and watchdog system reset
2	0x0002	INT0	External interrupt request 0
3	0x0004	INT1	External interrupt request 1
4	0x0006	PCINT0	Pin change interrupt request 0
5	0x0008	PCINT1	Pin change interrupt request 1
6	0x000A	PCINT2	Pin change interrupt request 2
7	0x000C	WDT	Watchdog time-out interrupt
8	0x000E	TIMER2 COMPA	Timer/Counter2 compare match A
9	0x0010	TIMER2 COMPB	Timer/Counter2 compare match B
10	0x0012	TIMER2 OVF	Timer/Counter2 overflow
11	0x0014	TIMER1 CAPT	Timer/Counter1 capture event
12	0x0016	TIMER1 COMPA	Timer/Counter1 compare match A
13	0x0018	TIMER1 COMPB	Timer/Counter1 compare match B
14	0x001A	TIMER1 OVF	Timer/Counter1 overflow
15	0x001C	TIMER0 COMPA	Timer/Counter0 compare match A
16	0x001E	TIMER0 COMPB	Timer/Counter0 compare match B
17	0x0020	TIMER0 OVF	Timer/Counter0 overflow
18	0x0022	SPI, STC	SPI serial transfer complete
19	0x0024	USART, RX	USART Rx complete
20	0x0026	USART, UDRE	USART, data register empty
21	0x0028	USART, TX	USART, Tx complete
22	0x002A	ADC	ADC conversion complete
23	0x002C	EE READY	EEPROM ready
24	0x002E	ANALOG COMP	Analog comparator
25	0x0030	TWI	2-wire serial interface
26	0x0032	SPM READY	Store program memory ready

H/W 인터럽트

4. Interrupt

(8) S/W 인터럽트?

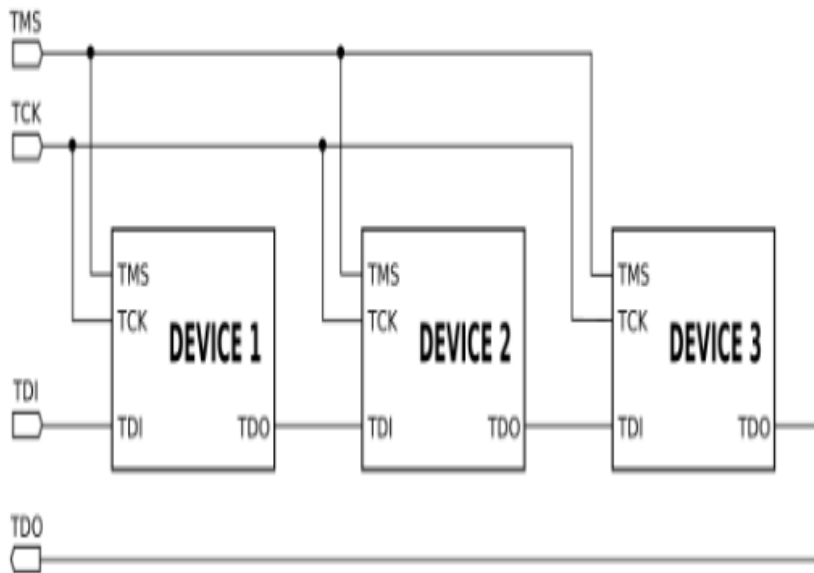
- CPU 내부의 명령어 처리 오류(Exception)나 시스템 호출(System call)에 의해 발생
- Exception : 명령어 실행 후 그 결과 instruction fault로 발생하며, 이벤트가 언제든지 예측 불가하게 발생하는 것이 아니라 **기준에 맞추어 또는 시간에 맞추어서 수행**시키는 것을 의미(**동기식 인터럽트**)
예로 CPU가 0으로 나누기, Page fault 가 발생한 경우 등이 있다.
- System Call : 응용프로그램에서 OS에 요청하는 SW 인터럽트
응용프로그램에서 인터럽트 요청 발생 -> C 라이브러리의 요청한 기능에 따른 시스템 호출 고유번호가 레지스터에 저장 -> 커널이 IDT(Interrupt Descriptor Table)을 참조하여 ISR을 수행

5. JTAG

(1) JTAG?

- 디지털 회로에서 디지털 입출력을 위해 **직렬통신 방식**으로 출력 데이터를 전송하거나 입력 데이터를 수신하는 방식
- 임베디드 시스템 개발 시에 사용하는 디버깅 장비 CPU의 기계어 코드를 실행하지 않고 내부 플래시 메모리나 CPU의 외부 플래시 메모리에 코드를 쓰거나 읽을 수 있는 장비가 대표적임

(2) JTAG H/W인터페이스



1. TDI (데이터 입력) : Test하기 위한 데이터 신호. TMS에 의해 전이된 TAP state에 따라, TDI가 command/data 가 결정됨
2. TDO (데이터 출력) : Test한 결과를 외부에서 모니터링 하기 위한 pin, 이 역시 TAP state에 따라 address/data가 될 수 있음.
3. TCK (클럭) : Test clock
4. TMS (모드) : Test Mode로 전환하기 위한 제어 신호
5. TRST (리셋)