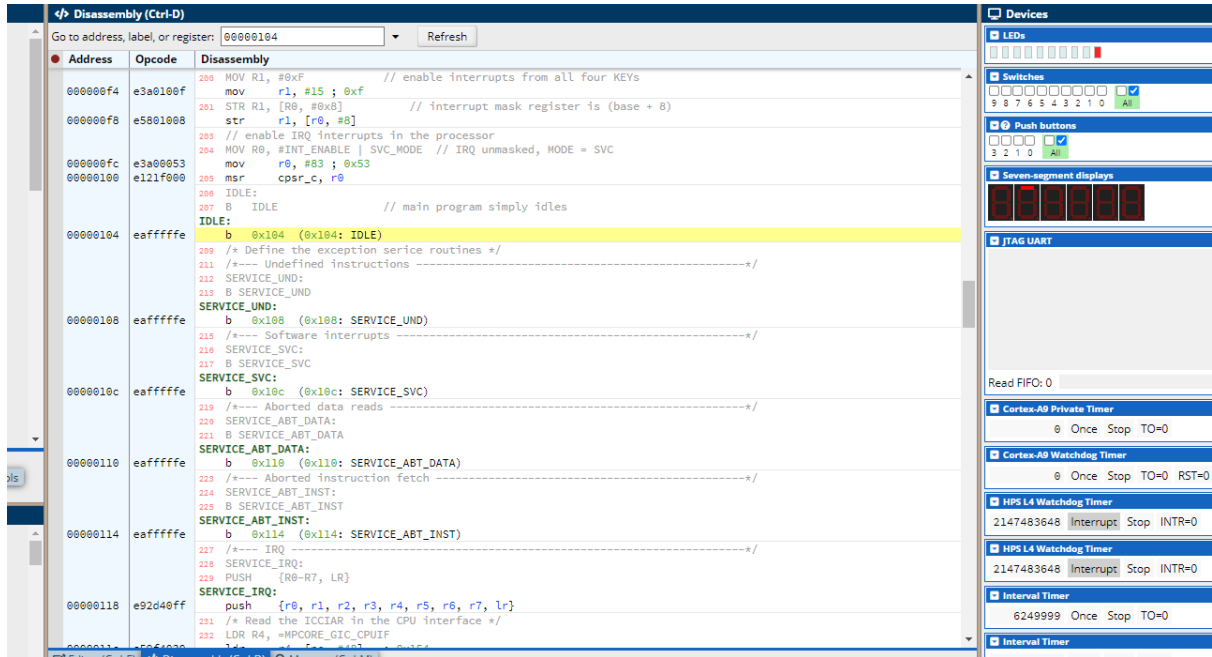


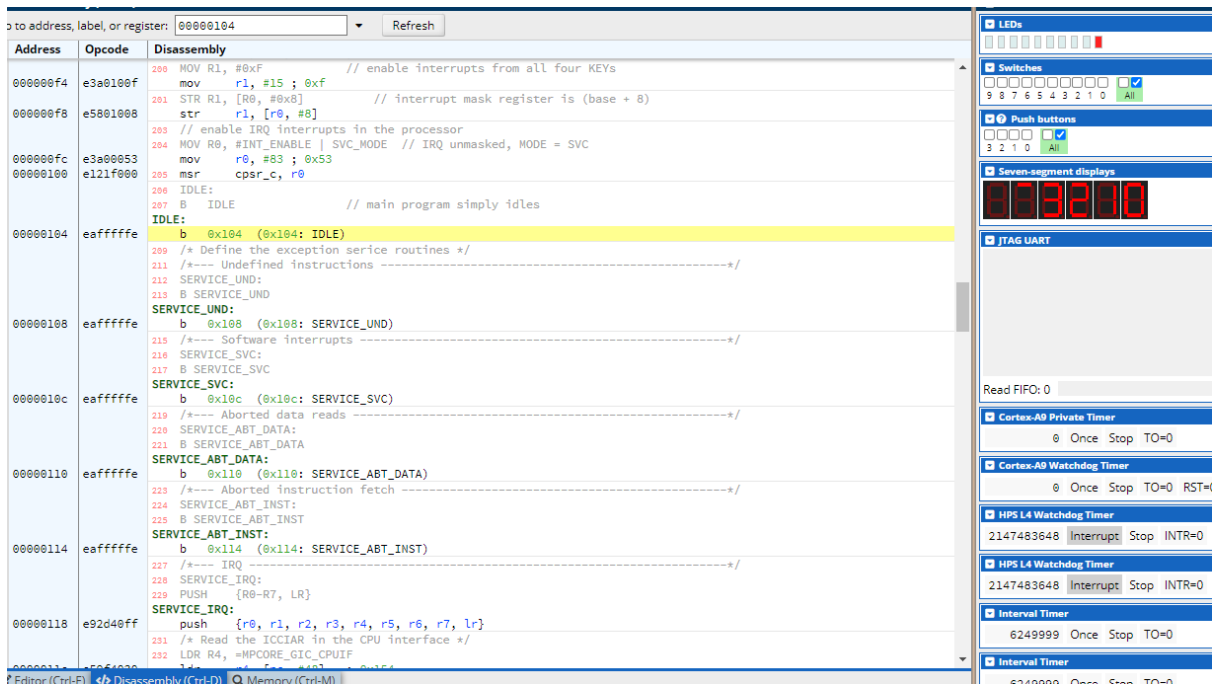
# 실습: DE1-SoC 인터럽트 프로그래밍

2017253019안희영

1. (어셈블리 언어 인터럽트 핸들러) 강의 자료와 별도 자료를 통해 제공된 인터럽트 핸들러 프로그램을 작성하여 시뮬레이터에서 실행시켜서 동작을 확인하시오. (시뮬레이터에서는 한 파일로 작성해야 함)



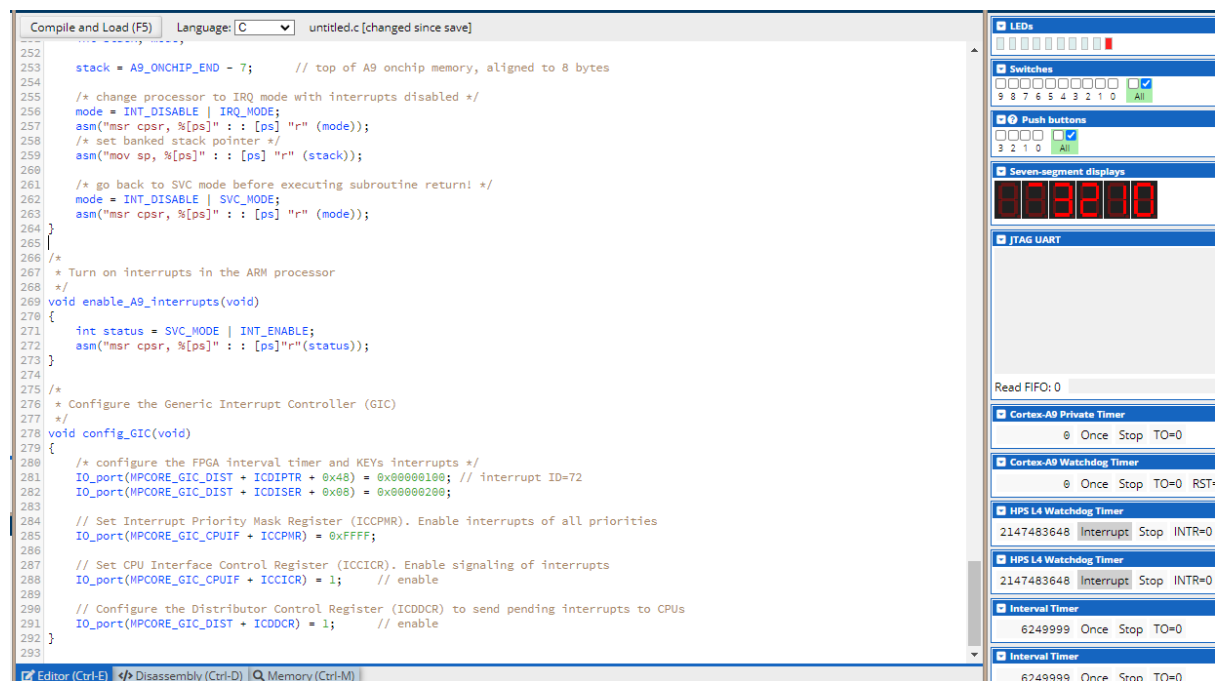
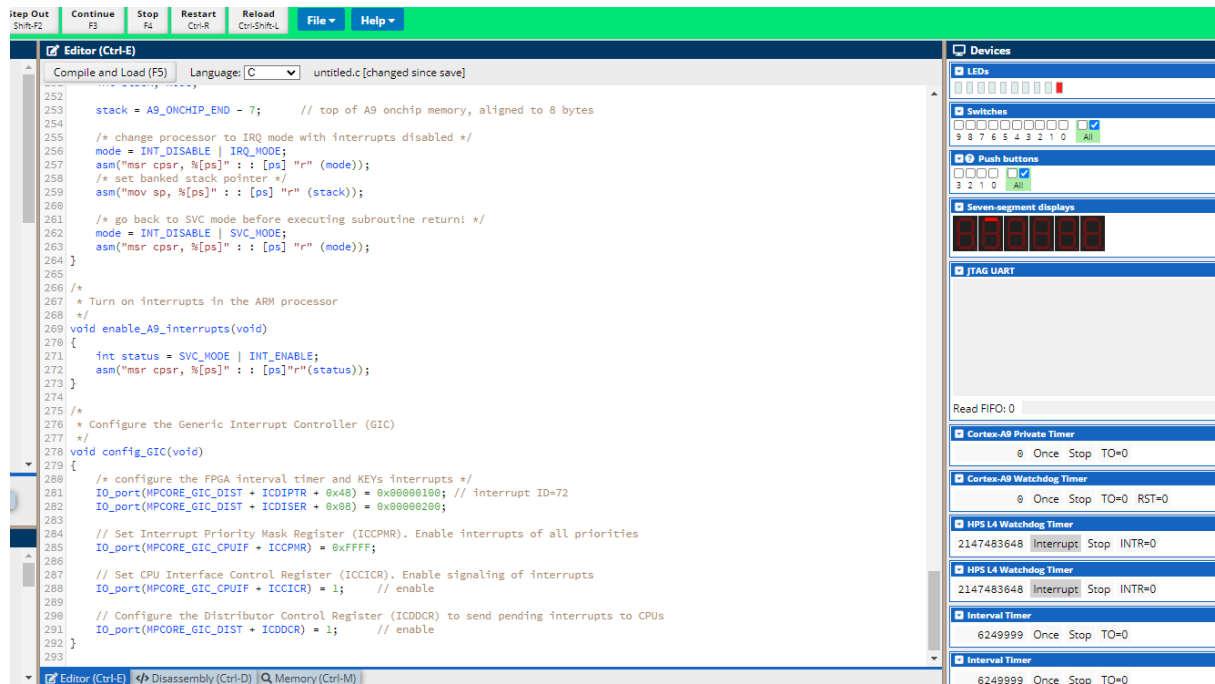
Address	Opcode	Disassembly
000000f4	e3a0100f	200 MOV R1, #0xf // enable interrupts from all four KEYS
000000f8	e5001008	201 STR R1, [R0, #0x8] // interrupt mask register is (base + 8)
000000fc	e3a00053	202 MOV R0, #INT_ENABLE   SVC_MODE // IRQ unmasked, MODE = SVC
00000100	e121f000	203 MSR cpsr_c, R0
00000104	ea000000	204 IDLE: // main program simply idles
00000108	ea000000	205 B IDLE
0000010c	ea000000	206 SERVICE_UND: // Define the exception service routines
00000110	ea000000	207 B SERVICE_UND
00000114	ea000000	208 SERVICE_SVC: // Aborted data reads
00000118	ea000000	209 B SERVICE_SVC



Address	Opcode	Disassembly
000000f4	e3a0100f	200 MOV R1, #0xf // enable interrupts from all four KEYS
000000f8	e5001008	201 STR R1, [R0, #0x8] // interrupt mask register is (base + 8)
000000fc	e3a00053	202 MOV R0, #INT_ENABLE   SVC_MODE // IRQ unmasked, MODE = SVC
00000100	e121f000	203 MSR cpsr_c, R0
00000104	ea000000	204 IDLE: // main program simply idles
00000108	ea000000	205 B IDLE
0000010c	ea000000	206 SERVICE_UND: // Define the exception service routines
00000110	ea000000	207 B SERVICE_UND
00000114	ea000000	208 SERVICE_SVC: // Aborted data reads
00000118	ea000000	209 B SERVICE_SVC

pushbutton을 눌렀다 때는순간 인터럽트 발생. 해당 인터럽트에 대해 hexled로 결과값이 나옴.

2. (C 언어 인터럽트 핸들러) 강의 자료와 별도 자료를 통해 제공된 인터럽트 핸들러 프로그램을 작성하여 실행시키고, 동작을 분석하시오.



interrupt\_ID.h : 인터럽트 ID 상수로 정의

```
#define KEY0      1
#define KEY1      2
#define KEY2      4
#define KEY3      8
- - - - -
```

키 입력 확인을 위한 비트 (1000, 0100, 0010, 0001) 상수로 선언

```

set_A9_IRQ_stack ();          // initialize the stack pointer for IRQ mode
config_GIC ();                // configure the general interrupt controller
config_KEYS ();               // configure pushbutton KEYS to generate interrupts
enable_A9_interrupts ();      // enable interrupts

IO_port(LED_R_BASE) = LED_pattern;
IO_port(HEX5_HEX4_BASE) = HEX_pattern;

while (1)
;
return 0;

```

set\_A9\_IRQ\_stack (): irq모드를 위한 스택포인터 설정

```

mode = INT_DISABLE | IRQ_MODE;
asm("msr cpsr, %[ps]" : : [ps] "r" (mode)); // irq 모드로 변경
asm("mov sp, %[ps]" : : [ps] "r" (stack)); // 스택 포인터 설정값으로 이동

mode = INT_DISABLE | SVC_MODE;
asm("msr cpsr, %[ps]" : : [ps] "r" (mode)); // svc 모드로 전환

```

config\_GIC (): gic 설정

```

IO_port(MPCORE_GIC_DIST + ICDIPTR + 0x48) = 0x00000100;
IO_port(MPCORE_GIC_DIST + ICDISER + 0x08) = 0x00000200; // 발생시 전달할 인터럽트 지정
IO_port(MPCORE_GIC_CPUIF + ICCPMR) = 0xFFFF; // 인터럽트 우선도 설정
IO_port(MPCORE_GIC_CPUIF + ICCICR) = 1; // 인터럽트를 전달하게 설정

```

IO\_port(MPCORE\_GIC\_DIST + ICDDCR) = 1; //배포자 활성화

config\_KEYS (): pushbutton 인터럽트 설정

IO\_port(KEY\_BASE + 8) = 0xF; // 모든 버튼 입력 인터럽트 활성화

enable\_A9\_interrupts: 인터럽트 enable 설정

asm("msr cpsr, %[ps]" : : [ps] "r" (status)); // svc 모드에서 인터럽트가 enable 상태가 되도록 or 연산 후 cpsr에 전달하여 설정함.

```

IO_port(LED_R_BASE) = LED_pattern; led 출력 주소에 1을 넣음
IO_port(HEX5_HEX4_BASE) = HEX_pattern; HEX5_HEX4_BASE 주소에 1을 할당.
while (1); // 인터럽트가 들어올 때까지 대기

void pushbutton_ISR( void ) // push 버튼 인터럽트 발생시 실행
int press, pattern;
press = IO_port(KEY_BASE + 0xC); // 에지 레지스터를 읽어옴
IO_port(KEY_BASE + 0xC) = press; // 인터럽트 초기화
key_hold ^= press; // 변화값을 key_hold에 저장
pattern = 0;
if (key_hold & KEY0) pattern = hex_conversions[0]; // 버튼 0이 눌린 경우 해당값 할당
if (key_hold & KEY1) pattern |= (hex_conversions[1]<<8); // 1이 눌린 경우 pattern에
// 쉬프트 연산하여 추가
if (key_hold & KEY2) pattern |= (hex_conversions[2]<<16); // key1과
// 같음 (쉬프트 16비트)

```

```
if (key_hold & KEY3) pattern |= (hex_conversions[3]<<24); // key1 과  
같은 (쉬프트 24 비트)  
IO_port(HEX3_HEX0_BASE) = pattern; // hex led의 출력주소에 전달_
```

3.

(1) Interrupt Processor Target Register (ICDIPTRn)의 용도에 대해서 적으시오.

각 인터럽트를 전달해야하는 CPU 인터페이스를 지정하는 데 사용

(2) 위의 레지스터에 대해서 interrupt ID 72와 interrupt ID 73의 target processor를 CPU0로 지정할 때에 어떻게 해야 하는지 말하시오.

0xFFFFD800의 바이트를 값 0x01로 설정

(3) Interrupt Set Enable Registers (ICDISERn)의 용도에 대해서 적으시오.

배포자에서 CPU 인터페이스로 지원 되는 각 인터럽트를 전달하는 데 사용

(4) 위의 레지스터에 대해서 interrupt ID 72와 interrupt ID 73의 forwarding을 enable하려면 어떻게 해야 하는지 말하시오. (5) GIC에서 인터럽트를 발생시킨 peripheral의 interrupt ID를 어떻게 알 수 있는지 말하시오.

ICDISERn를 0x00000300으로 세트

4. Pushbutton Key Port에 대해서 다음 물음에 답하시오. (DE1-SoC Computer 컴퓨터자료, p 10 및 p21 참조)

(1) Interrupt mask register의 기능은?

입력 포트에 있는 입력 데이터 값이 변경될 때 GIC에 인터럽트 신호를 보낼지 여부를 지정

(2) Edgecapture register의 기능은?

입력 데이터가 마지막으로 읽은 이후에 0에서 1로 변경되는 경우에 에지캡처 레지스터의 해당 비트 위치가 1의 값을 갖는다.

(3) 처리된 pushbutton 인터럽트를 clear 하려면 어떻게 해야 하는가?

KEY\_ptr + 3의 위치에 에지캡처 레지스터값을 전달한다.