adc
advanced digital chips inc

Preliminary

CANTUS

– *INTERRUPT* –

32bits EISC Microprocessor *CANTUS*

Ver 1.0
October 8, 2009

Advanced Digital Chips Inc.

History

2009-10-08	Created Preliminary Specification

CANTUS Evaluation Board Application Note : #0003 INTERRUPT

Office

8th Floor, KookMin 1 Bldg.,

1009-5, Daechi-Dong, Gangnam-Gu, Seoul, 135-280, Korea

Tel : + 82-2-2107-5800

Fax : + 82-2-571-4890

URL : http://www.adc.co.kr

— Table of Contents —

— List of Figures —

— List of Tables —

1 Summary

이 문서는 CANTUS의 INTERRUPT에 대한 Application Note이다.

CANTUS는 32개 채널의 INTERRUPT 입력을 가지며, 이 입력들은 Timer, SPI, TWI, UART 등과 같은 내부 장치에서 발생하는 30개의 인터럽트와 외부 2개의 인터럽트로 구성된다.

2 Register Set

2.1 Register Set Flow Chart

CANTUS의 Interrupt를 사용하기 위해선 다음과 같은 순서로 Register를 설정한다.

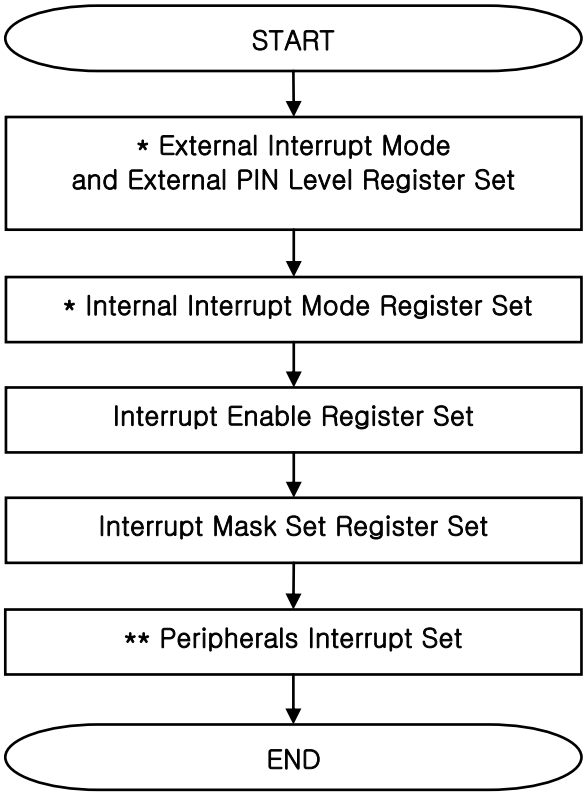

그림 2-1 Register Set Flow Chart

* - 필요할 경우 설정
** - 사용하는 Peripheral에 따라 설정해 주어야 함.

Interrupt가 발생하여 ISR에서 처리한 후 Interrupt Pending Clear Register를 통해 해당 Interrupt Vector No. 값으로 Clear 하여야 한다. (SDK의 interrupt.c는 이를 수행한 후 setinterrupt() 함수로 등록한 사용자ISR을 수행한다.)

2.2 External Interrupt Mode and External PIN Level Register

외부 Interrupt를 사용할 경우 아래와 같이 설정 한다.

표 2-1 External Interrupt Mode and External PIN Level Register (EINTMOD)

Address : 0x8002_0804

Bit	***R/W***	***Description***	***Default Value***
31 : 8	R	Reserved	-
7	R	EIRQ1ST : EIRQ1 PIN Level	-
6 : 4	R/W	EIRQ1MOD : EIRQ1 Active State 000 : Low Level 001 : High Level 010 : Falling Edge 011 : Rising Edge 1xx : Any Edge	010
3	R	EIRQ0ST : EIRQ0 PIN Level	-
2 : 0	R/W	EIRQ0MOD : EIRQ0 Active State 000 : Low Level 001 : High Level 010 : Falling Edge 011 : Rising Edge 1xx : Any Edge	010

2.3 Internal Interrupt Mode Register

표 2-2 Internal Interrupt Mode Register (IINTMOD)

Address : 0x8002_0808

Bit	*R/W*	*Description*	*Default Value*
31	R/W	Vector No. 0x3F Interrupt Mode bit	1
30	R/W	Vector No. 0x3E Interrupt Mode bit	1
29	R/W	Vector No. 0x3D Interrupt Mode bit	1
28	R/W	Vector No. 0x3C Interrupt Mode bit	1
27	R/W	Vector No. 0x3B Interrupt Mode bit	1
26	R/W	Vector No. 0x3A Interrupt Mode bit	1
25	R/W	Vector No. 0x39 Interrupt Mode bit	1
24	R/W	Vector No. 0x38 Interrupt Mode bit	1
23	R/W	Vector No. 0x37 Interrupt Mode bit	1
22	R/W	Vector No. 0x36 Interrupt Mode bit	1
21	R/W	Vector No. 0x35 Interrupt Mode bit	1
20	R/W	Vector No. 0x34 Interrupt Mode bit	1
19	R/W	Vector No. 0x33 Interrupt Mode bit	1
18	R/W	Vector No. 0x32 Interrupt Mode bit	1
17	R/W	Vector No. 0x31 Interrupt Mode bit	1
16	R/W	Vector No. 0x30 Interrupt Mode bit	1
15	R/W	Vector No. 0x2F Interrupt Mode bit	1
14	R/W	Vector No. 0x2E Interrupt Mode bit	1
13	R/W	Vector No. 0x2D Interrupt Mode bit	1
12	R/W	Vector No. 0x2C Interrupt Mode bit	1
11	R/W	Vector No. 0x2B Interrupt Mode bit	1
10	R/W	Vector No. 0x2A Interrupt Mode bit	1
9	R/W	Vector No. 0x29 Interrupt Mode bit	1
8	R/W	Vector No. 0x28 Interrupt Mode bit	1
7	R/W	Vector No. 0x27 Interrupt Mode bit	1
6	R/W	Vector No. 0x26 Interrupt Mode bit	1
5	R/W	Vector No. 0x25 Interrupt Mode bit	1
4	-	Reserved	-
3	R/W	Vector No. 0x23 Interrupt Mode bit	1
2	R/W	Vector No. 0x22 Interrupt Mode bit	1
1	R/W	Vector No. 0x21 Interrupt Mode bit	1
0	-	Reserved	-

*** Internal Interrupt Mode bit
0 : High Level Mode
1 : Rising Edge Mode

2.4 Interrupt Enable Register

Interrupt Enable Register가 Set되지 않으면 Interrupt Pending Register에 값을 저장할 수 없다.

표 2-3 Interrupt Enable Register (INTEN)

Address : 0x8002_0810

Bit	*R/W*	*Description*	*Default Value*
31	R/W	Vector No. 0x3F Interrupt Enable bit	0
30	R/W	Vector No. 0x3E Interrupt Enable bit	0
29	R/W	Vector No. 0x3D Interrupt Enable bit	0
28	R/W	Vector No. 0x3C Interrupt Enable bit	0
27	R/W	Vector No. 0x3B Interrupt Enable bit	0
26	R/W	Vector No. 0x3A Interrupt Enable bit	0
25	R/W	Vector No. 0x39 Interrupt Enable bit	0
24	R/W	Vector No. 0x38 Interrupt Enable bit	0
23	R/W	Vector No. 0x37 Interrupt Enable bit	0
22	R/W	Vector No. 0x36 Interrupt Enable bit	0
21	R/W	Vector No. 0x35 Interrupt Enable bit	0
20	R/W	Vector No. 0x34 Interrupt Enable bit	0
19	R/W	Vector No. 0x33 Interrupt Enable bit	0
18	R/W	Vector No. 0x32 Interrupt Enable bit	0
17	R/W	Vector No. 0x31 Interrupt Enable bit	0
16	R/W	Vector No. 0x30 Interrupt Enable bit	0
15	R/W	Vector No. 0x2F Interrupt Enable bit	0
14	R/W	Vector No. 0x2E Interrupt Enable bit	0
13	R/W	Vector No. 0x2D Interrupt Enable bit	0
12	R/W	Vector No. 0x2C Interrupt Enable bit	0
11	R/W	Vector No. 0x2B Interrupt Enable bit	0
10	R/W	Vector No. 0x2A Interrupt Enable bit	0
9	R/W	Vector No. 0x29 Interrupt Enable bit	0
8	R/W	Vector No. 0x28 Interrupt Enable bit	0
7	R/W	Vector No. 0x27 Interrupt Enable bit	0
6	R/W	Vector No. 0x26 Interrupt Enable bit	0
5	R/W	Vector No. 0x25 Interrupt Enable bit	0
4	R/W	Vector No. 0x24 Interrupt Enable bit	0
3	R/W	Vector No. 0x23 Interrupt Enable bit	0
2	R/W	Vector No. 0x22 Interrupt Enable bit	0
1	R/W	Vector No. 0x21 Interrupt Enable bit	0
0	R/W	Vector No. 0x20 Interrupt Enable bit	0

*** Interrupt Enable bit
0 : Interrupt Disable
1 : Interrupt Enable

2.5 Interrupt Mask Set Register

Pending Register에 값이 저장되어도 Interrupt Mask Set Register가 Set되지 않으면 CPU에 Interrupt 요청을 할 수 없다.

표 2-4 Interrupt Mask Set Register (MASKSET)

Address : 0x8002_0814

Bit	*R/W*	*Description*	*Default Value*
31	W	Vector No. 0x3F Interrupt Request Set bit	0
30	W	Vector No. 0x3E Interrupt Request Set bit	0
29	W	Vector No. 0x3D Interrupt Request Set bit	0
28	W	Vector No. 0x3C Interrupt Request Set bit	0
27	W	Vector No. 0x3B Interrupt Request Set bit	0
26	W	Vector No. 0x3A Interrupt Request Set bit	0
25	W	Vector No. 0x39 Interrupt Request Set bit	0
24	W	Vector No. 0x38 Interrupt Request Set bit	0
23	W	Vector No. 0x37 Interrupt Request Set bit	0
22	W	Vector No. 0x36 Interrupt Request Set bit	0
21	W	Vector No. 0x35 Interrupt Request Set bit	0
20	W	Vector No. 0x34 Interrupt Request Set bit	0
19	W	Vector No. 0x33 Interrupt Request Set bit	0
18	W	Vector No. 0x32 Interrupt Request Set bit	0
17	W	Vector No. 0x31 Interrupt Request Set bit	0
16	W	Vector No. 0x30 Interrupt Request Set bit	0
15	W	Vector No. 0x2F Interrupt Request Set bit	0
14	W	Vector No. 0x2E Interrupt Request Set bit	0
13	W	Vector No. 0x2D Interrupt Request Set bit	0
12	W	Vector No. 0x2C Interrupt Request Set bit	0
11	W	Vector No. 0x2B Interrupt Request Set bit	0
10	W	Vector No. 0x2A Interrupt Request Set bit	0
9	W	Vector No. 0x29 Interrupt Request Set bit	0
8	W	Vector No. 0x28 Interrupt Request Set bit	0
7	W	Vector No. 0x27 Interrupt Request Set bit	0
6	W	Vector No. 0x26 Interrupt Request Set bit	0
5	W	Vector No. 0x25 Interrupt Request Set bit	0
4	W	Vector No. 0x24 Interrupt Request Set bit	0
3	W	Vector No. 0x23 Interrupt Request Set bit	0
2	W	Vector No. 0x22 Interrupt Request Set bit	0
1	W	Vector No. 0x21 Interrupt Request Set bit	0
0	W	Vector No. 0x20 Interrupt Request Set bit	0

*** Interrupt Request Set bit
0 : No Effect interrupt Mask.
1 : Pending interrupt is allowed to become active (interrupts sent to CPU).

2.6 Peripherals Interrupt Set

CANTUS의 Peripheral 중 Interrupt를 사용하기 위해선 Peripheral Register의 Interrupt 관련 Set이 필요한 것이 있다.

3 Function Set

3.1 InitInterrupt()

매개 변수 없음.

\ Cantuslib \ Interrupt.c

SDK의 예제는 기본적으로 Interrupt.c의 InitInterrupt() 함수를 호출한다.

```
void InitInterrupt()
{
   SETVECTORED;
   INTERRUPT_ENABLE;

   *R_INTEN = 0; //all disable
   *R_INTMASKCLR = 0xffffffff; // all interrupt disable
   //set vector base register
   asm("push %r0");
   asm("ldi        _vector_table,%r0"); //program load address
   asm("mvtc       0,%r12");
   asm("sync");
   asm("pop %r0");
}
```

이 함수는 CANTUS의 Interrupt를 사용하기 위한 기본 설정(status register set[1])과 vector base address를 설정한다. 사용자가 Interrupt를 사용하지 않더라도, 이 함수를 호출하여야 한다. 호출하지 않을 경우 Exception을 처리 할 수 없다.

[1] Core ISA Reference Manual(Ae32000-isa-rm_ko.pdf) 참조

3.2 setinterrupt()

BOOL setinterrupt(INTERRUPT_TYPE intnum,void (*fp)());

- Return : Interrupt.h에 정의되지 않은 Interrupt_type이면 FALSE
- INTERRUPT_TYPE intnum : 사용할 Interrupt type (\include \interrupt.h)
- Void (*fp)() : ISR로 사용할 함수

```
typedef enum
{
        INTNUM_EIRQ0=0,
        INTNUM_TIMER0,
        INTNUM_I2S,
        INTNUM_UART0,

        INTNUM_EIRQ1,//4
        INTNUM_TIMER1,
        INTNUM_DMA0,
        INTNUM_UART1,

        INTNUM_GPIOA,//8
        INTNUM_TIMER2,
        INTNUM_USB,
        INTNUM_UART2,

        INTNUM_GPIOB,//0xc
        INTNUM_TIMER3,
        INTNUM_SPI,
        INTNUM_UART3,

        INTNUM_GPIOC,//0x10
        INTNUM_TIMER4,
        INTNUM_TWI_RTC,
        INTNUM_UART4_OCR2A,

        INTNUM_GPIOD,//0x14
        INTNUM_TIMER5,
        INTNUM_VOICE,
        INTNUM_UART5_OCR2B,

        INTNUM_GPIOE,//0x18
        INTNUM_TIMER6,
        INTNUM_NFCTRL_SDHC,
        INTNUM_UART6_OCR3A,

        INTNUM_GPIOF,//0x1c
        INTNUM_TIMER7_KEYSCAN,
        INTNUM_DMA1,
        INTNUM_UART7_OCR3B,

        INTNUM_MAX
}INTERRUPT_TYPE;
```

```
static void (*UserVector_table[32])(void) =
{
   NOTUSEDISR,
   NOTUSEDISR,
   NOTUSEDISR,
   NOTUSEDISR,
   NOTUSEDISR,
   NOTUSEDISR,
   NOTUSEDISR,
   NOTUSEDISR,
   NOTUSEDISR,
   NOTUSEDISR,
   NOTUSEDISR,
   NOTUSEDISR,
   NOTUSEDISR,
   NOTUSEDISR,
   NOTUSEDISR,
   NOTUSEDISR,
   NOTUSEDISR,
   NOTUSEDISR,
   NOTUSEDISR,
   NOTUSEDISR,
   NOTUSEDISR,
   NOTUSEDISR,
   NOTUSEDISR,
   NOTUSEDISR,
   NOTUSEDISR,
   NOTUSEDISR,
   NOTUSEDISR,
   NOTUSEDISR,
   NOTUSEDISR,
   NOTUSEDISR,
   NOTUSEDISR,
   NOTUSEDISR,
};
BOOL setinterrupt(INTERRUPT_TYPE intnum,void (*fp)())
{
    if (intnum >= INTNUM_MAX)
        return FALSE;

    UserVector_table[intnum]=fp;
    return TRUE;
}
```

CANTUS의 Interrupt를 사용하기 위해 사용자는 ISR(Interrupt Service Routine)을 정의하여야 한다. setinterrupt() 함수는 사용자가 사용할 INTTERUPT_TYPE과 ISR로 사용할 함수를 인자로 넘겨줌으로써 INTERRUPT_TYPE의 ISR로 정의하여 사용할 수 있다.

ISR을 호출하는 과정은 다음과 같이 이루어 진다.

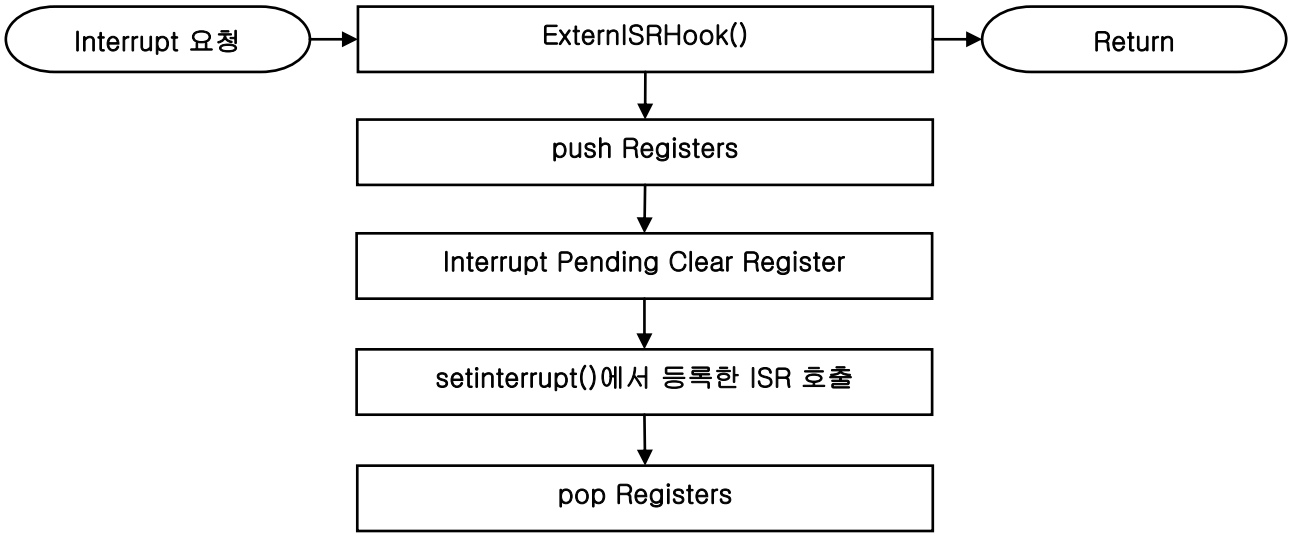

그림 3-1 ISR 호출 과정

3.3 EnableInterrupt()

void EnableInterrupt(INTERRUPT_TYPE num,BOOL b);

- INTERRUPT_TYPE num : Interrupt type (\ include \ interrupt.h)
- BOOL b : TRUE – ENABLE / FALSE –DISABLE

```
void EnableInterrupt(INTERRUPT_TYPE num,BOOL b)
{
   CRITICAL_ENTER();
   if (!b) //disable
   {
      *R_INTEN &= (~(1 << num));
      *R_INTMASKCLR |= ( 1 << num);
   }
   else
   {
      *R_INTMASKSET = (1 << num);
      *R_INTEN |= ( 1 << num);
   }
   CRITICAL_EXIT();
}
```

사용할 ISR을 등록하였다면 Interrupt를 Enable 함으로써 해당 ISR을 사용할 수 있다.
EnableInterrupt() 함수는 해당 Interrupt를 Enable/Disable 시킨다.

3.4 Interrupt.c

CANTUS의 Interrupt를 사용하기 위한 정의 및 함수로 구성되어 있다. 사용자는 Interrupt.c의 함수를 사용하여 편리하게 Interrupt를 설정 할 수 있다.

CANTUS의 Interrupt의 자세한 내용은 아래 문서를 참조하라.

- CANTUS Datasheet 9 INTERRUPTS
- Core ISA Reference Manual(AE32000-isa-rm-ko.pdf) 2.5 Exceptions
- EISC Software Develop Guide(EISC_Software_Developer_Guide_v2_0.pdf) 5장 Exception