

Forum: [16-bit PIC24/dsPIC](#)  
Topic: Early Interrupt的問題  
Subject: Early Interrupt的問題  
作者: hj411335  
2020年05月21日 15:05:43

最近在研究dspic33CK系列的ADC專章的early interrupt，看完說明與專章的範例(example 5.5)之後仍不太瞭解

Early interrupt是在conversion ready之前就進入ADC ISR，而在範例裡面又ISR裡面直接讀取ADC buffer。但當下應該還沒conversion ready，請問這樣讀取會得到即時的ADC數值嗎？

以下是專章example

```
// These variables will keep the conversion result.
volatile unsigned short dataANO; // dedicated core
volatile unsigned short dataAN2; // shared core
int main()
{
// ADC INITIALIZATION
// Configure the I/O pins to be used as analog inputs.
ANSELAbits.ANSA0 = 1; TRISAbits.TRISA0 = 1; // AN0/RA0 connected the dedicated core 0
ANSELAbits.ANSA2 = 1; TRISAbits.TRISA2 = 1; // AN2/RA2 connected the shared core
// Configure the common ADC clock.
ADCON3Hbits.CLKSEL = 2; // clock from FRC oscillator
ADCON3Hbits.CLKDIV = 0; // no clock divider (1:1)
// Configure the cores' ADC clock.
ADCORE0Hbits.ADCS = 0; // dedicated core clock divider (1:2)
ADCON2Lbits.SHRADCS = 0; // shared core clock divider (1:2)
// Configure sample time for shared core.
ADCON2Hbits.SHRSAMC = 10; // 12 TAD sample time
// Configure the ADC reference sources.
ADCON3Lbits.REFSEL = 0; // AVdd as voltage reference
// Configure the integer of fractional output format.
ADCON1Hbits.FORM = 0; // integer format
// Select single-ended input configuration and unsigned output format.
ADMOD0Lbits.SIGNO = 0; // AN0/RA0
ADMOD0Lbits.DIFF0 = 0; // AN0/RA0
ADMOD0Lbits.SIGN2 = 0; // AN2/RA2
// Enable and calibrate the module.
EnableAndCalibrate(); // See Example 5-1
// Configure and enable early ADC interrupts.
ADCORE0Hbits.EISEL = 0; // early interrupt is generated 1 TADCORE clock prior
// to when the data is ready
ADCORE1Hbits.EISEL = 3; // early interrupt is generated 4 TADCORE clocks prior
// to when the data is ready
ADCON2Lbits.EIEN = 1; // enable early interrupts for ALL inputs
```

```

ADEIELbits.EIEN0 = 1; // enable interrupt for AN0
ADEIELbits.EIEN2 = 1; // enable interrupt for AN2
_ADCAN0IF = 0; // clear interrupt flag for AN0
_ADCAN0IE = 1; // enable interrupt for AN0
_ADCAN2IF = 0; // clear interrupt flag for AN2
_ADCAN2IE = 1; // enable interrupt for AN2
// Set same trigger source for all inputs to sample signals simultaneously.
ADTRIG0Lbits.TRGSR0 = 13; // timer 2 for AN0
ADTRIG0Hbits.TRGSR2 = 13; // timer 2 for AN2
// TIMER 2 INITIALIZATION (TIMER IS USED AS A TRIGGER SOURCE FOR ALL CHANNELS).
T2CONbits.TCS = 0; // clock from peripheral clock
T2CONbits.TCKPS = 0; // 1:1 prescale
PR2 = 0x8000; // rollover every 0x8000 clocks
T2CONbits.TON = 1; // start timer to generate ADC triggers
while(1);
return 1;
}
// ADC AN0 ISR (DEDICATED CORE)
void __attribute__((interrupt, no_auto_psv)) _ADCAN0Interrupt(void)
{
dataAN0 = ADCBUF0; // read conversion result
_ADCAN0IF = 0; // clear interrupt flag
}
// ADC AN2 ISR (SHARED CORE)
void __attribute__((interrupt, no_auto_psv)) _ADCAN2Interrupt(void)
{
dataAN2 = ADCBUF2; // read conversion result
_ADCAN2IF = 0; // clear interrupt flag
}

```