Application Note

32-bit Cortex™-M0 MCU
NuMicro® Family

How to use Analog-Digital converter at single mode?

Application Note

Table of Contents-

1 INTRODUCTION

NUC1xx family contains one 12-bit successive approximation analog-to-digital converters (SAR A/D converter) with 8 input channels. The A/D converter supports three operation modes: single, single-cycle scan and continuous scan mode. There are two kinds of scan mode: continuous mode and single cycle mode. The A/D converters can be started by software and external STADC/PB.8 pin. The details of feature and structure are as follows.

Note that the analog input port pins must be configured as input type before ADC function is enabled.

1.1 Features

- Analog input voltage range: 0~Vref (Max to 5.0V).
- 12-bits resolution and 10-bits accuracy is guaranteed.
- Up to 8 single-end analog input channels or 4 differential analog input channels.
- Maximum ADC clock frequency is 16MHz.
- Up to 800K SPS conversion rate, conversion time is less than 1.25us.
- Three operating modes
- Single mode: Single channel A/D conversion
- Single-cycle scan mode: Continuous conversion on enabled channels
- Continuous scan mode: Repetitive conversion on enabled channels
- An A/D conversion can be started by
- Software write 1 to ADST bit
- External pin STADC
- Conversion results are held in data registers for each channel with valid and overrun indicators.
- Conversion result can be compared with specify value and user can select whether to generate an interrupt when conversion result is equal to the compare register setting.
- Channel 7 support 3 input source: external analog voltage, internal fixed bandgap voltage and internal temperature sensor output.
- Support Self-calibration to minimum conversion error.

1.2 Structure

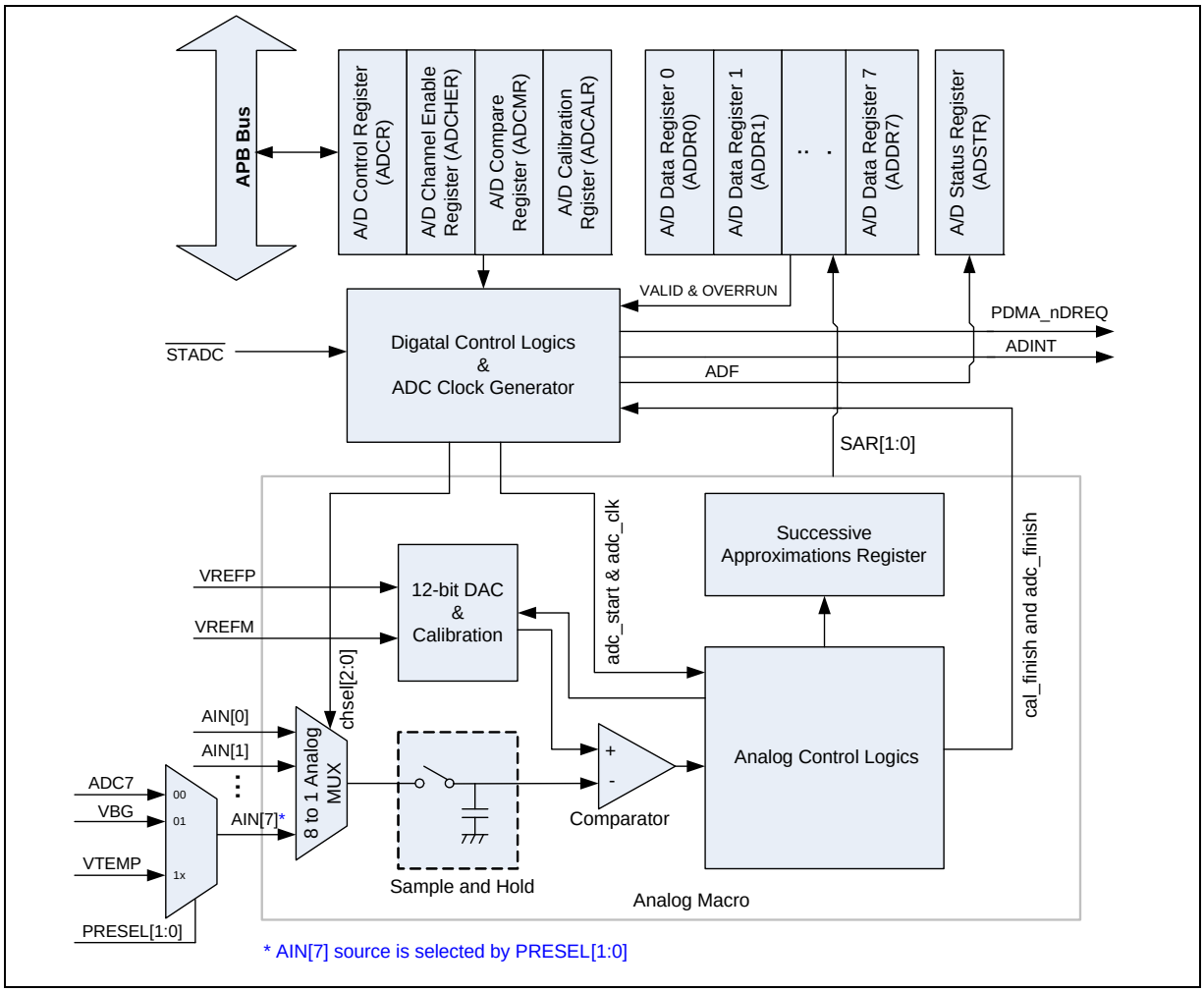

Figure 1 ADC Controller Block Diagram

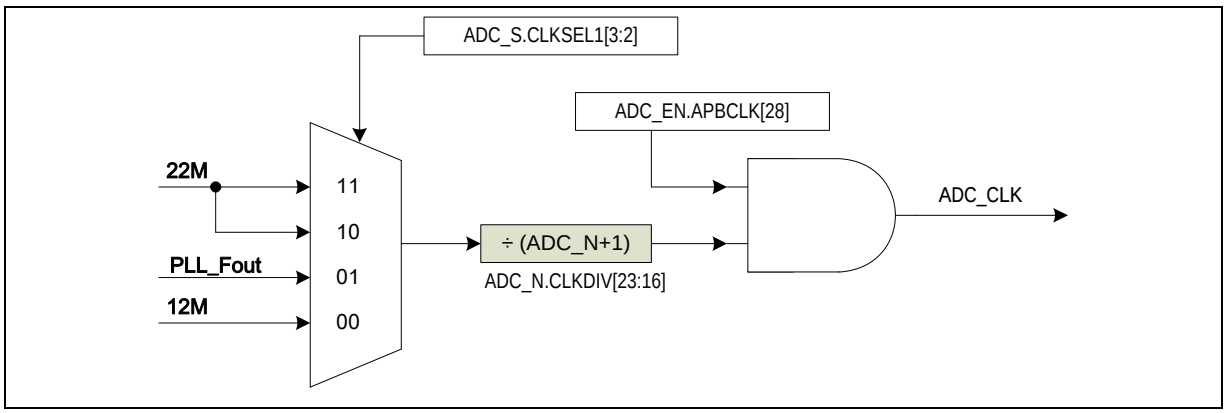

Figure 2 ADC Clock Control

1.3 Operation Procedure

The A/D converter operates by successive approximation with 12-bit resolution. This A/D converter equips with self calibration function to minimum conversion error, user can write 1 to CALEN bit in ADCALR register to enable calibration function, while internal calibration is finish the CAL_DONE bit will assert. The ADC has two operation modes: single mode and scan mode. There are two kinds of scan mode: continuous mode and single-cycle mode. When changing the operating mode or analog input channel enable, in order to prevent incorrect operation, software must clear ADST bit to 0 in ADCR register. The A/D converter discards current conversion immediately and enters idle state while ADST bit is cleared.

1.3.1 Self-Calibration

User can write 1 to CALEN bit in ADCALR register to enable self calibration. The operation is process internally and it needs 127 ADC clocks to complete calibration. After CALEN is set to 1, software must wait CAL_DONE bit set by internal hardware. The detail timing is shown as below:

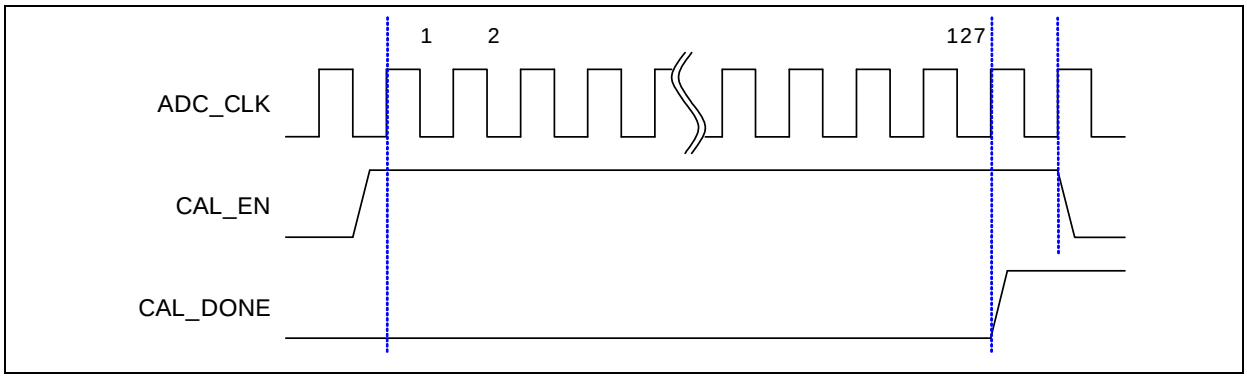

Figure 3 ADC Converter Self-Calibration Timing Diagram

1.3.2 ADC Clock Generator

The maximum sampling rate is up to 800KHz and the conversion time is less 1.25us. It needs 20 ADC clocks to complete an A/D conversion. The ADC engine has three clock source selected by 2-bit ADC_S(CLKSEL[3:2]), the ADC clock frequency is divided by an 8-bit prescaler with the formula:

The ADC clock frequency = (ADC clock source frequency) / (ADC_N+1);
where the 8-bit ADC_N is located in register CLKDIV[23:16].

In generally, software can set ADC_S and ADC_N to get 16MHZ or slightly less.

1.3.3 Single Mode

In single mode, A/D conversion is to be performed only once on the specified single channel. The operations are as follows:

1. A/D conversion is started when the ADST bit in ADCR is set to 1 by software or external trigger input.
2. When A/D conversion is finished, the result is transferred to the A/D data register corresponding to the channel.

3. On completion of conversion, the ADF bit in ADSR is set to 1. If the ADIE bit is set to 1 at this time, an ADINT interrupt request is generated.

4. The ADST bit remains 1 during A/D conversion. When A/D conversion ends, the ADST bit is automatically cleared to 0 and the A/D converter enters in idle state. If the ADST bit is cleared to 0 during A/D conversion, A/D conversion will stop immediately and enter in idle state.

Note: If software enables more than one channel in single mode, the least channel is converted and other enabled channels will be ignored.

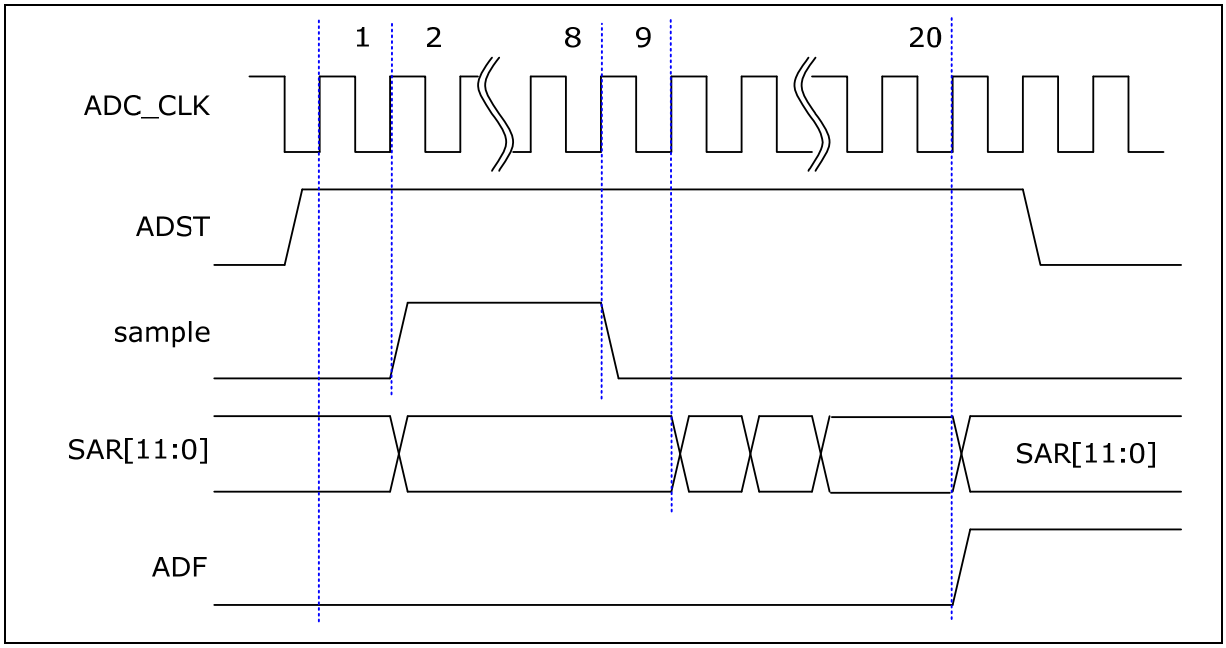

Figure 4 Single Mode Conversion Timing Diagram

1.3.4 Continuous Scan Mode

In continuous scan mode, A/D conversion is to be performed sequentially on the specified channels that enabled by CHEN bits in ADCHER register (maximum 8 channels for ADC). The operations are as follows:

1. When the ADST bit in ADCR is set to 1 by software or external trigger input, A/D conversion starts on the channel with the lowest number.

2. When A/D conversion for each enabled channel is completed, the result is sequentially transferred to the A/D data register corresponding to each enabled channel.

3. When conversion of all the selected channels that is enabled is completed, the ADF bit in ADSR is set to 1. If the ADIE bit is set to 1 at this time, an ADINT interrupt is requested after A/D conversion ends. Conversion of the 1st enabled channel starts again.

4. Steps 2 to 3 are repeated as long as the ADST bit remains set to 1. When the ADST bit is cleared to 0, A/D conversion stops and the A/D converter enters the idle state.

An example timing diagram for continuous scan on enabled channels (0, 2, 3 and 7) is shown as

below:

(This example is only appropriate for ADC)

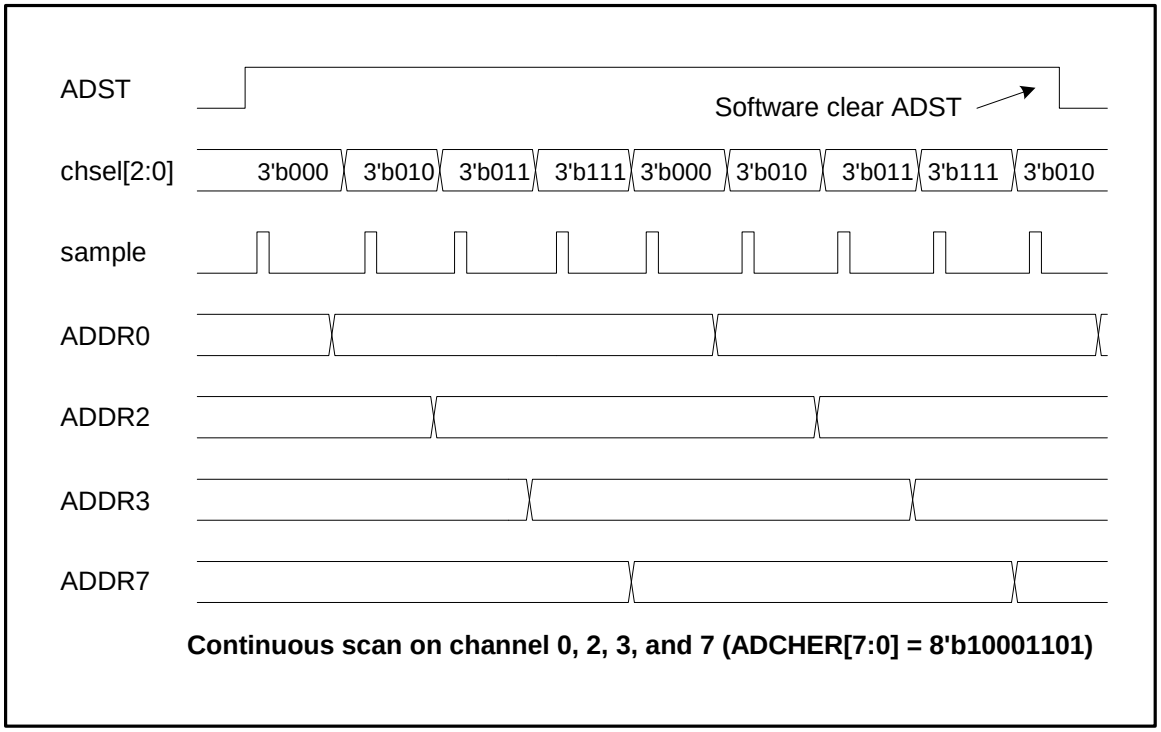

Figure 5 Continuous Scan on Enabled Channels Timing Diagram

1.3.5 Single-Cycle Scan Mode

In single-cycle scan mode, A/D conversion will sample and convert the specified channels once in the sequence from the least to highest channel. Operations are as follows:

1. When the ADST bit in ADCR is set to 1 by a software or external trigger input, A/D conversion starts on the channel with the lowest number.
2. When A/D conversion for each enabled channel is completed, the result is sequentially transferred to the A/D data register corresponding to each channel.
3. When conversion of all the selected channels that is enabled is completed, the ADF bit in ADSR is set to 1. If the ADIE bit is set to 1 at this time, an ADINT interrupt is requested after A/D conversion ends.
4. After A/D conversion ends, the ADST bit is automatically cleared to 0 and the A/D converter enters in idle state. If the ADST bit is cleared to 0 during A/D conversion, A/D conversion will stop immediately and enter in idle state.

An example timing diagram for single-cycle scan on enabled channels (0, 2, 3 and 7) is shown as below:

(This example is only appropriate for ADC)

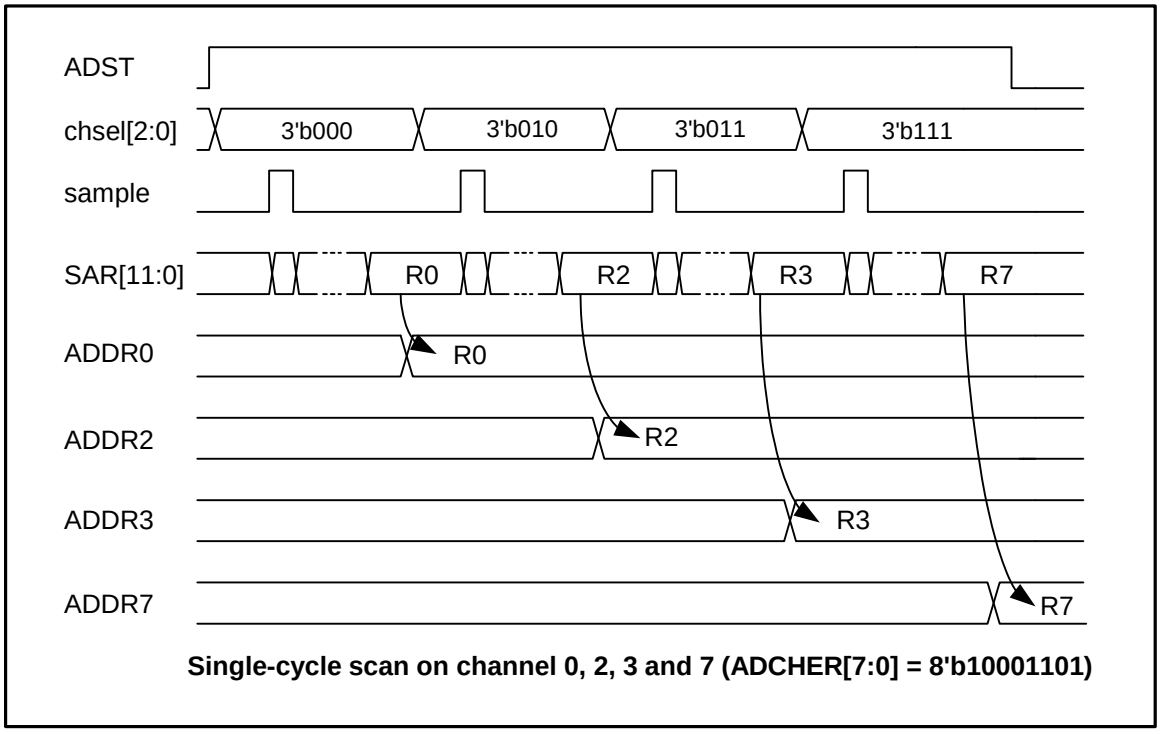

Figure 6 Single-Cycle Scan on Enabled Channels Timing Diagram

1.3.6 Input Sampling and A/D Conversion Time

The A/D converter has a built-in sample-and-hold circuit that samples the analog input when A/D conversion start delay time (Td) has passed after ADST bit in ADCR is set to 1, then start conversion. Due to ADC clock is generated by PCLK divided by (N+1), the maximum delay time from APB write to A/D start sampling analog input time is 2N PCLKs. The start delay time is shown as below:

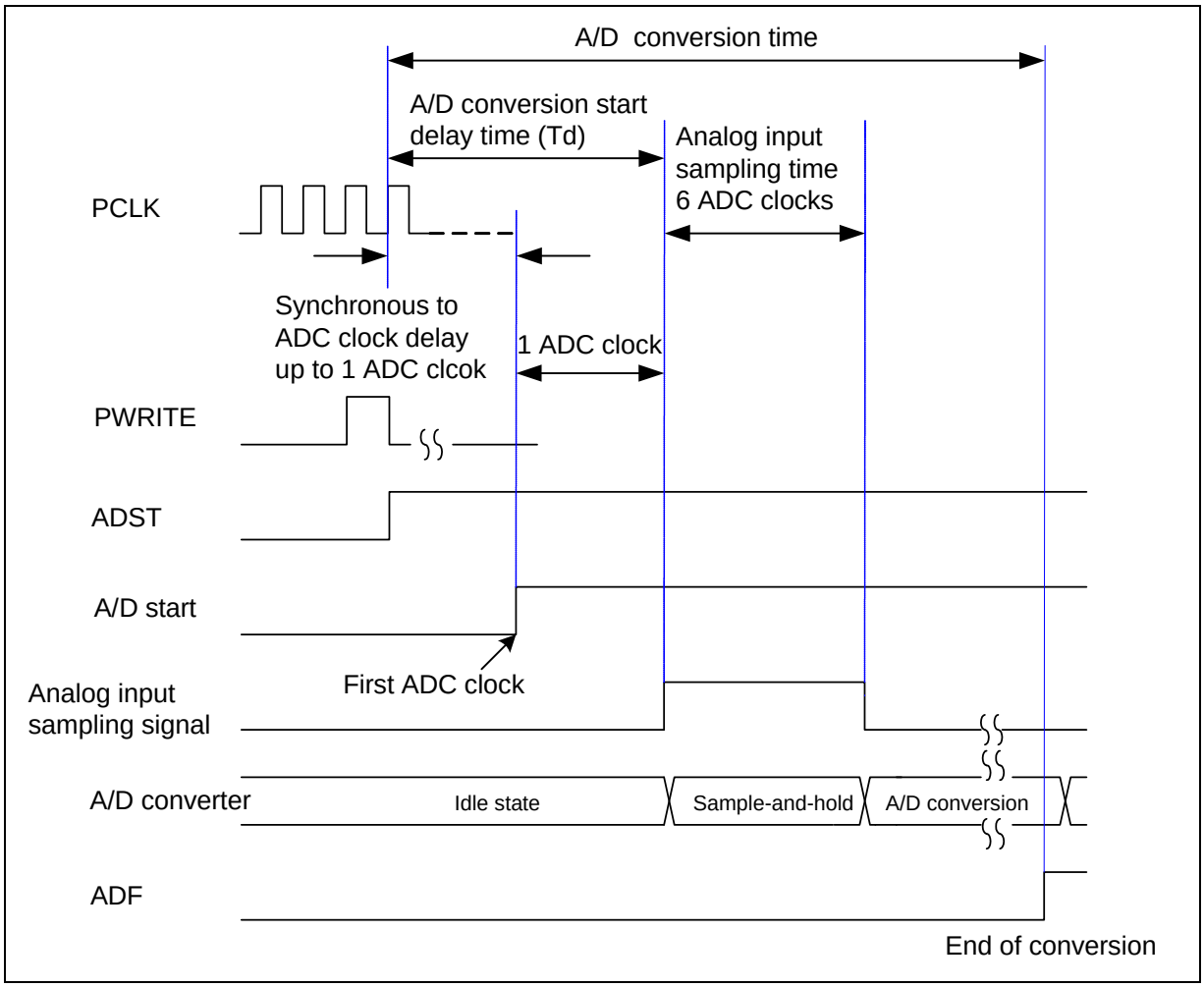

Figure 7 Conversion Start Delay Timing Diagram

A/D conversion can be triggered by external pin request. When the ADCR.TRGEN is set to high to enable ADC external trigger function, setting the TRGS[1:0] bits to 2'b00 is to select external trigger input from the STADC pin. Software can set TRGCOND[1:0] to select trigger condition is **falling/rising edge or low/high** level. An 8-bit sampling counter is used to deglitch. If level trigger condition is selected, the STADC pin must be kept at defined state at least 8 PCLKs. The ADST bit will be set to 1 at the 9th PCLK and start to conversion. Conversion is continuous if external trigger input is pull at low (or high state) in level trigger mode. It is stopped only when external condition trigger condition disappears. If edge trigger condition is selected, the high and low state must be kept at least 4 PLCKs. Pulse that is shorter than this specification will be ignored. Other operations, in both single and scan modes, are the same as when the ADST bit has been set to 1.

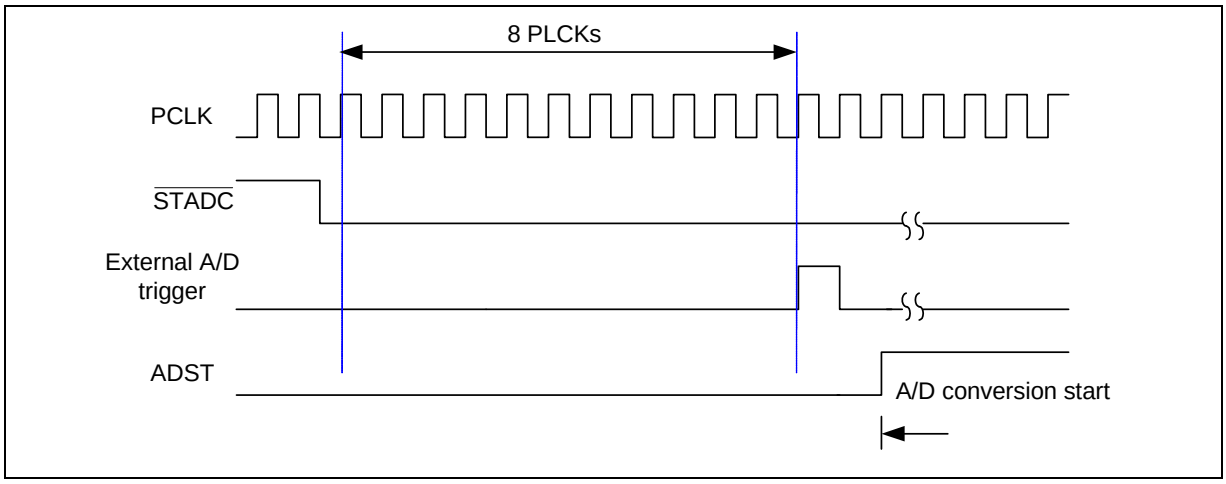

Figure 8 External A/D Conversion Trigger Timing Diagram

1.3.7 Conversion Result Monitor by Compare Mode

NUC1XX controller provide two sets of compare register ADCMPR0 and 1 to monitor maximum two specified channels conversion result from A/D conversion module, refer to Figure 9. Software can select which channel to be monitored by set CMPCH[2:0] and CMPCOND bit is used to check conversion result is less than specify value or greater than (equal to) value specified in CMPD[11:0]. When the compare result meets the setting, compare match counter will increase 1, when counter value reach the setting of CMPMATCNT then CMPF bit will be set to 1, if CMPIE bit is set then an ADINT interrupt request is generated. Software can use it to monitor the external analog input pin voltage transition in scan mode without imposing a load on software. Detail logics diagram is shown as below:

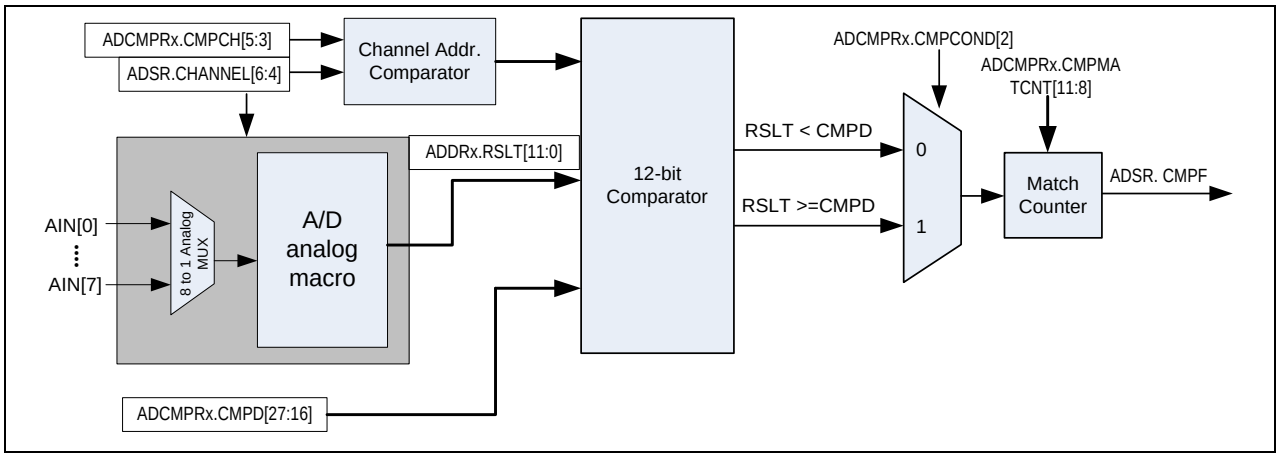

Figure 9 A/D Conversion Result Monitor Logics Diagram

1.3.8 Interrupt Sources

The A/D converter generates a conversion end ADF in ADSR register upon the end of A/D conversion. If ADIE bit in ADCR is set then conversion end interrupt request ADINT is generated. If CMPIE bit is enabled, when A/D conversion result meets setting in ADCMPR register, monitor interrupt is

generated, ADINT will be set also. CPU can clear CMPF and ADF to stop interrupt request.

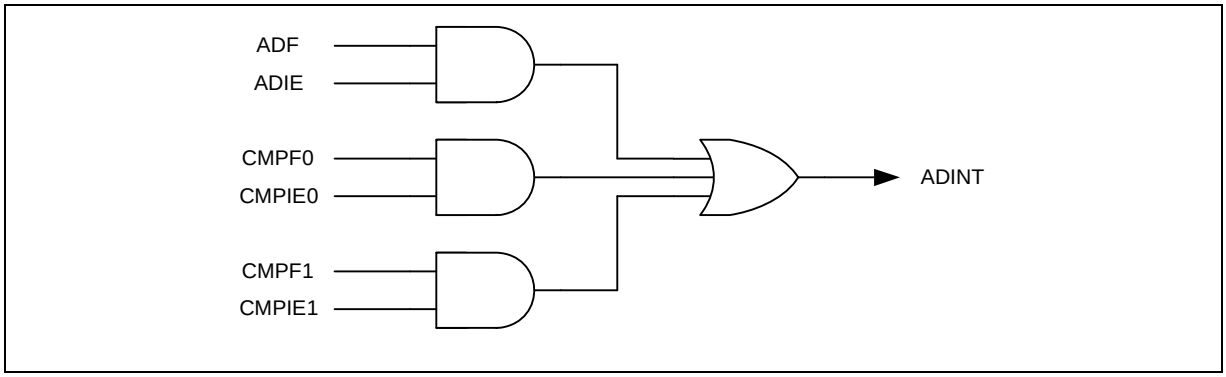

Figure 10 A/D Controller Interrupt

1.3.9 Peripheral DMA Request

When A/D conversion is finished converted result is loaded into ADDR register and VALID bit is set to 1. If PTEN bit in ACDR is set, ADC controller will generate PDMA request (P_nDRQ) to ask data transfer. Having the converted result read by PDMA in response to P_nDRQ enables continuous conversion to be achieved without intervening CPU.

2 HOW TO PROGRAM ADC

2.1 PROGRAM FLOW OF ADC for single mode

1. Set analog input mode of GPIOA selected for ADC function.
2. Select **ADC_S** bit in **CLKSEL1** register and **ADC_N** bit in **CLKDIV** register and then set the **ADC_EN** bit in **APBCLK** register to enable **ADC** clock source
3. Set ADEN bit in **ADCR** register and select **DIFF** bit in **ADCR** register and **ADMD** bit in **ADCR** register for operation mode
4. Select **CHEN** bit in **ADCHER** register to define transferred channel
5. Set **ADIE** bit in **ADCR** register and **ISER** to enable interrupt.
6. Set **ADST** bit in **ADCR** register to start conversion

2.2 Sample code

```
#include <stdio.h>
#include "NUC1xx.h"

/*---------------------------------------------------------------------------
  Define variable
  ---------------------------------------------------------------------------*/
            static uint32_t ADCTemp=0;
/*---------------------------------------------------------------------------
  Function subroutine
  ---------------------------------------------------------------------------*/
void Delay(uint32_t delayCnt)
{
    while(delayCnt--)
    {
        __NOP();
        __NOP();
    }
}
/*---------------------------------------------------------------------------
  Interrupt subroutine
  ---------------------------------------------------------------------------*/
void ADC_IRQHandler(void) // Timer0 interrupt subroutine
{
            ADC->ADSR.ADF=1;
```

```
        ADCTemp=ADC->ADDR[1].RSLT;
}
/*---------------------------------------------------------------------------
  MAIN function
  ---------------------------------------------------------------------------*/
int32_t main (void)
{

        NVIC_DisableIRQ(ADC_IRQn); //Disable ADC interrupt
        outpw(&ADC->ADCR ,0 );              //Disable ADC

        /* Step 1. GPIO initial */
        GPIOA->PMD.PMD1=0;                  //Set input mode
        GPIOA->SCH|=0x00020000;             //Disable digital input path
        SYS->GPAMFP.ADC1=1;                 //Set ADC function

        /* Step 2. Enable and Select ADC clock source, and then enable ADC module */
        SYSCLK->CLKSEL1.ADC_S = 2;          //Select 22Mhz for ADC
        SYSCLK->CLKDIV.ADC_N = 1;           //ADC clock source = 22Mhz/2 =11Mhz;
        SYSCLK->APBCLK.ADC_EN = 1;          //Enable clock source

        /* Step 3. Select Operation mode */
        ADC->ADCR.ADEN = 1;                 //Enable ADC module
        ADC->ADCR.DIFF = 0;                 //single end input
        ADC->ADCR.ADMD = 0;                 //single mode

        /* Step 4. Select ADC channel */
        ADC->ADCHER.CHEN = 0x02;

        /* Step 5. Enable ADC interrupt */
        ADC->ADSR.ADF =1;                   //clear the A/D interrupt flags for safe
        ADC->ADCR.ADIE = 1;
        NVIC_EnableIRQ(ADC_IRQn);

        /* Step 6. Conversion start  */
```

```
        ADC->ADCR.ADST=1;
        while(1)
        {
                if (ADC->ADSR.BUSY==0) ADC->ADCR.ADST=1;
                ADCTemp=ADC->ADSR.CHANNEL;
        }
}
```

3 REVISION HISTORY

REV.	DATE	DESCRIPTION
1.00	March 1, 2010	1. Initially issued.
1.01	April 8, 2010	1. Delete register description

Important Notice

Nuvoton products are not designed, intended, authorized or warranted for use as components in systems or equipment intended for surgical implantation, atomic energy control instruments, airplane or spaceship instruments, transportation instruments, traffic signal instruments, combustion control instruments, or for other applications intended to support or sustain life. Further more, Nuvoton products are not intended for applications wherein failure of Nuvoton products could result or lead to a situation wherein personal injury, death or severe property or environmental damage could occur.

Nuvoton customers using or selling these products for use in such applications do so at their own risk and agree to fully indemnify Nuvoton for any damages resulting from such improper use or sales.

Please note that all data and specifications are subject to change without notice. All the trademarks of products and companies mentioned in this datasheet belong to their respective owners.