Freescale Semiconductor
Application Note

Document Number: AN3949
Rev. 1, 04/2010

ADC16 Calibration Procedure and Programmable Delay Block Synchronization

For the MCF51EM256

by: Inga Harris
Applications Engineer
East Kilbride, U.K.

Contents

1 Introduction

The MCF51EM256 ColdFire MCU has an integrated 16-bit ADC and a programmable delay block (PDB). The integration of these modules is not standardized across different devices. The device integration of these modules is customized for each MCU family to better suit the needs of the targeted application.

This document describes the procedure required to perform the analog-to-digital converter (ADC) auto-calibration function and provides software that may be used as a template for application code development. The second section of this application note describes the integration of the ADC16 and PDB peripherals for the MCF51EM256 MCU to complement the block-specific information provided in the reference manual. It also describes how to use these modules together in a simple application and synchronize them.

NOTE

With the exception of mask set errata documents, if any other Freescale document contains information that conflicts with the information in the device reference manual; the reference manual must be considered to have the most current and correct data.

2 ADC16 Calibration

The 16-bit ADC requires self-calibration to improve the ADCs linearity and to ensure that the specified accuracies in the data sheet are met. This calibration must be executed at least once after a power-on. The calibration registers are in volatile memory with the results then stored in the flash, or once after every reset if the results are not stored in the flash as shown in the accompanying software. The calibration frequency is a trade-off between accuracy and ADC overhead. This calibration must be executed to generate the offset and gain compensation values. These values are automatically subtracted (offset) and scaled (gain) during the conversion sequence to compensate for linearity errors in the SAR's internal DAC output as illustrated in Figure 1. The offset registers are also user-configurable for custom offset.

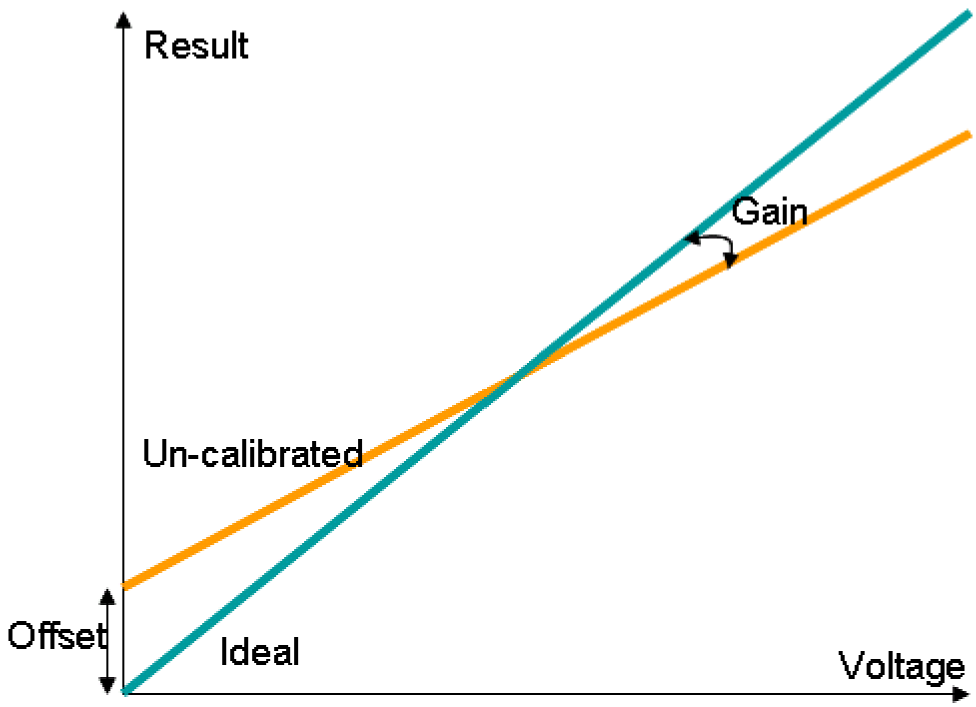

Figure 1. Gain and offset effect on result

Calibration is a three step process:

1. Configure the ADC
2. Initiate the hardware calibration and wait for the COCO flag
3. Generate gain compensation values

The values in the offset register are left-justified two's-complement values. Offset is subtracted from the right-justified successive-approximation conversion result before the compensated value is shifted into the

ADC result register. The offset value can be adjusted by the user. However, the adjusted calibration offset value must be stored in the secured memory. The un-calibrated default offset value is zero.

There are two gain registers, Plus-Side (ADCnPGH:L) and Minus-Side (ADCnMGH:L). They are used to compensate for a non-ideal output response of the internal DACs on each side of the differential ADC input. The Plus-Side value is used in single-ended mode and on the Plus-Side input of the differential inputs. The Minus-Side is used only in differential modes.

The values in the 16-bit gain registers represent integers from a recommended minimum value of 1.0 (0 x 8000), which is a full sample to a maximum allowed value of 1.03125 (0 x 83FF), with the decimal point fixed between bit 14 and 15 of the 16-bit register. The un-calibrated default register value is 1.0239 (0 x 8310). The input voltage is scaled by the appropriate gain compensation value, the positive input is scaled by the Plus Gain register (ADCnPGH:L), and the negative input is scaled by the Minus Gain register (ADCnMGH:L). See Equation 1 for the single ended mode and Equation 2 for the differential mode.

Single ended:

$$Code_{out} = 2^N\left[\left(\frac{V_{in}}{V_{REFH} - V_{REFL}}\right)\left(\frac{Cs_p}{Cn_p}\right)\right]$$

Eqn. 1

Differential:

$$Code_{out} = 2^N\left[\left(\frac{V_{in_p}Cs_p}{Cn_p} - \frac{V_{in_n}Cs_n}{Cn_n}\right)\left(\frac{1}{V_{REFH} - V_{REFL}}\right)\right]$$

Eqn. 2

Where Cs_x is the sampled capacitor, Cn_x is the nominal capacitor, and x is p-positive or n-negative.

2.1 Calibration Flow

Figure 2 shows the calibration process flow chart.

The calibration results can be affected by the clock source, frequency, power and conversion speed settings, voltage reference, hardware average function, the sample time and to a much lesser extent, environment. To achieve the accuracies specified in the electrical specifications the ADC clock frequency must be between 2 MHz and 4 MHz. The ADC must be set for high speed mode with ADLPC = 0, ADHSC = 1, 32x averaging set by setting AVGE = 1, and AVGS = 0 x 11. It is recommended that calibration be run with both VDDA and VREFH at or above 3 V. The input channel, conversion mode setting, compare function values, resolution, and differential and single-ended settings have no effect on the calibration result. However, they can be configured prior to calibration to later save time for re-configuring the ADC, post-calibraion.

The ADTRG bit in ADCSC2 must be cleared to enable the conversion initiation by the software trigger before initiating the calibration sequence by setting the CAL bit in ADCSC3. After the calibration sequence has been initiated the software must wait for the conversion complete bit (COCO) to be set.

The CALF is set and the CAL bit cleared if any ADC registers are written during the calibration sequence, if the stop mode is entered during the calibration sequence, or if the CAL bit is set while the ADC is in hardware trigger mode. The application code must be written to avoid or at least manage these scenarios. In the software example in AN3949SW, the calibration error flag is cleared and the code moves on completing the function with the failed calibration results.

After the ADC has completed its auto-calibration, the application code must complete the calibration procedure by calculating the gain compensation values for the plus and minus side DACs. This is executed by initializing a 16-bit area of RAM. The Plus Side calibration results are to be added together (CLP0, CLP1, CLP2, CLP3, CLP4, and CLPS) and stored in the RAM. This number is then divided by two and the MSB set. This 16-bit result can then be stored in the Plus Side Gain register ADCPGH:L. This routine is then repeated for the Minus Side registers CLM0, CLM1, CLM2, CLM3, CLM4, and CLMS. The result stored in ADCMGH:L, especially if differential inputs are used. The example code in AN3949SW is written in C to demonstrate the intension. However, for increased efficiency, the above routine can be written in assembly. The gain calculation is achieved efficiently by setting the carry bit and rotating-right through the carry bit on the high byte and again on the low byte.

Further calibrations can be initiated by clearing and then setting the CAL bit in ADCSC3. To allow for repeated calls to the example calibration function, the CAL bit can be cleared prior to exit. However, this may leave the application open to unwanted calibrations in the event of the CAL bit being written unintentionally.

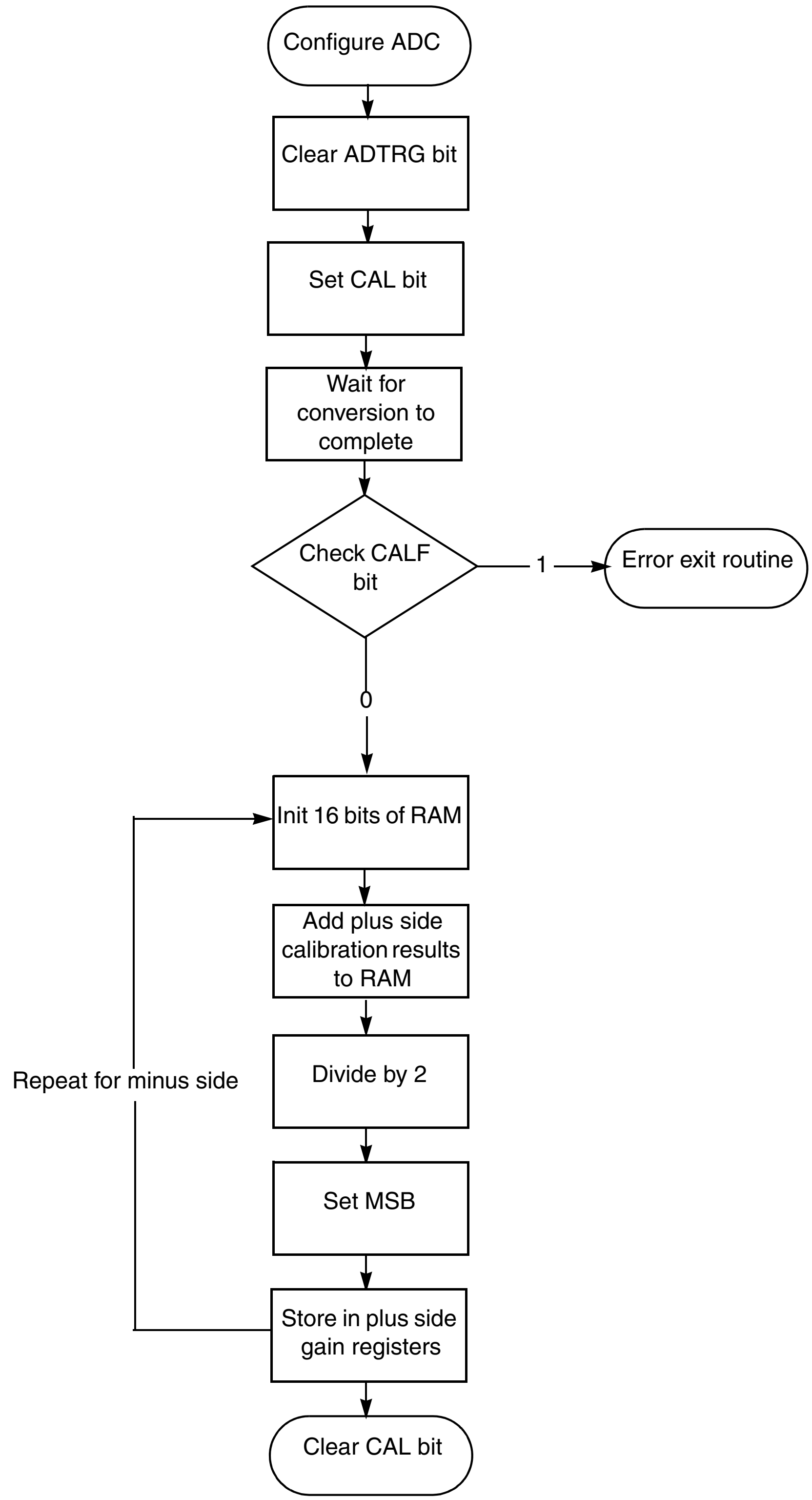

Figure 2. Calibration flow chart for the ADC16 module

2.2 Calibration Latency

The calibration routine may take as many as 15,000 ADCK cycles plus 100 bus cycles. To reduce this overhead, the calibration values, offset, plus and minus side gain, and plus and minus side calibration values, can be stored in the secured flash (non-volatile memory) by the application code after the initial calibration. This reduces the calibration overhead to 20 register store operations on subsequent POR, internal reset, and stop2 mode recoveries.

Other ways to reduce the time taken to run calibration are:

- Reduce amount of hardware averaging used (this has an effect on the accuracy)
- Disable interrupts to avoid calibration fails and get results as soon as possible
- Use the described assembly instructions (setting the carry bit and rotating-right through the carry bit)
- When doing single ended calibrations, the Minus Side Gain register computation can be skipped.

3 ADC16 and Programmable Delay Block (PDB) Integration

Figure 3 shows a simplified representation of the integration of the PDB and the 16-bit ADC on the MCF51EM256 with clocks and registers removed.

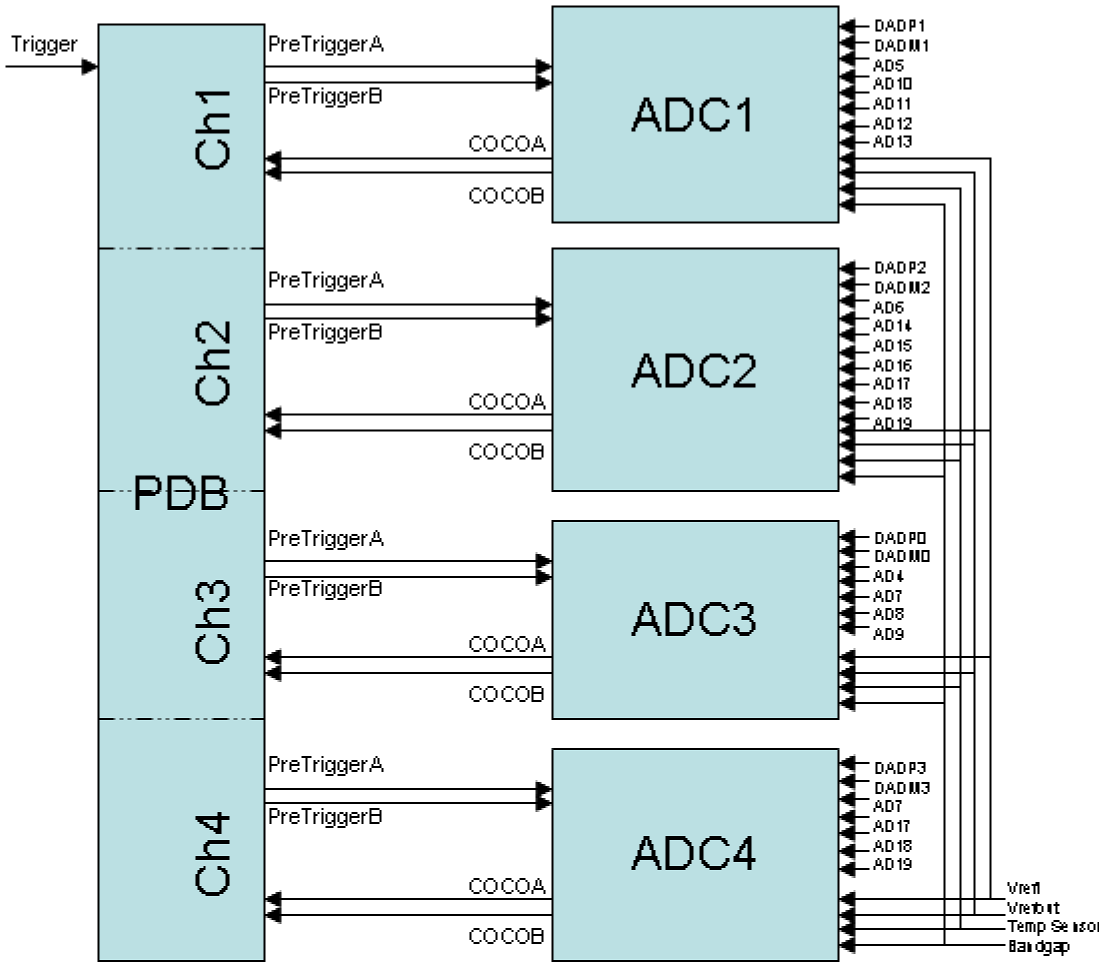

Figure 3. PDB and ADC integration on the MCF51EM256

This integration has been optimized for use in metering applications. Energy metering applications take two measurements for every reading (voltage and current) at regular intervals (due to fundamental frequency of mains supply) with only one interrupt per cycle. By using the PDB, hardware triggers can be sent to the ADC to initiate conversions at pre-set time intervals for detailed control of ADC conversion timing. Each ADC module has two status and control 1 registers (ADCnSC1A:B) and two result registers (ADCRA:B) that correspond to a PDB trigger derived from one event and the two delay timings, CHnDELA and CHnDELB.

Table 1 shows how the PDB channels, pre-triggers, and ADC hardware trigger signals correspond on the MCF51EM256.

Table 1. PDB to ADC correspondence

PDB Channel Pre-Trigger	ADC Trigger Select
PDBCH1 PreTriggerA	ADC1HWTSA
PDBCH1 PreTriggerB	ADC1HWTSB
PDBCH2 PreTriggerA	ADC2HWTSA
PDBCH2 PreTriggerB	ADC2HWTSB

Table 1. PDB to ADC correspondence (continued)

PDBCH3 PreTriggerA	ADC3HWTSA
PDBCH3 PreTriggerB	ADC3HWTSB
PDBCH4 PreTriggerA	ADC4HWTSA
PDBCH4 PreTriggerB	ADC4HWTSB

Every ADC has associated differential input, temperature sensor, bandgap, Vrefo, and Vrefl channels. The single ended channels are also associated with different ADCs. Table 2 is a detailed table of the MCF51EM256 ADC channel assignments.

Table 2. ADC channel assignments

Channel	ADC1	ADC2	ADC3	ADC4
1	Vrefl	Vrefl	DAD0	Vrefl
2	DAD1	Vrefl	Vrefl	Vrefl
3	Vrefl	DAD2	Vrefl	Vrefl
4	Vrefl	Vrefl	AD4	DAD3
5	AD5	Vrefl	Vrefl	Vrefl
6	Vrefl	AD6	Vrefl	Vrefl
7	Vrefl	Vrefl	Vrefl	AD7
8	Vrefl	Vrefl	AD8	Vrefl
9	Vrefl	Vrefl	AD9	Vrefl
10	AD10	Vrefl	Vrefl	Vrefl
11	AD11	Vrefl	Vrefl	Vrefl
12	AD12	Vrefl	Vrefl	Vrefl
13	AD13	Vrefl	AD13	Vrefl
14	Vrefl	AD14	Vrefl	Vrefl
15	Vrefl	AD15	Vrefl	Vrefl
16	Vrefl	AD16	Vrefl	Vrefl
17	Vrefl	AD17	Vrefl	AD17
18	Vrefl	AD18	Vrefl	AD18
19	Vrefl	AD19	Vrefl	AD19
20	Vrefout	Vrefout	Vrefout	Vrefout
21	Vrefl	N/A	N/A	N/A
22-25	N/A	N/A	N/A	N/A
26	TempSensor	TempSensor	TempSensor	TempSensor
27	Bandgap	Bandgap	Bandgap	Bandgap

CAUTION

Regardless of whether the application requires the RTC module or not the RTC and tamper must be initialized to stop interrupts and resets. There is a finite amount of time to do this. It must be executed before any application functions are called (in the initialization code).

```
// Disable write protection sequence 00 – 01 – 11 – 10
IRTC_CTRL = 0;//0b00;
IRTC_CTRL = 1;//0b01;
IRTC_CTRL = 3;//0b11;
IRTC_CTRL = 2;//0b10;

// RTC - Disable All RTC interrupts including tamper
IRTC_IER = 0;
IRTC_CTRL = 15<<9;   // maximum Tamper duration
IRTC_CTRL |= 0x02;   // Enable Write protect
```

The RTC control bit field (iRTC_CTRL) must run a specific sequence "key" before the interrupts can be disabled. Tamper duration can then be maximized and the write protection enabled.

3.1 Trigger Timings

The MCF51EM256 series ADC block contains duplicate control and result registers, allowing them to operate in a ping-pong fashion, for aexample A-B-A-B-A... and so on, alternating conversions between two different analog sources (per converter) as shown in Figure 4. The pre-trigger signals are used to indicate which ADC channel is sampled next.

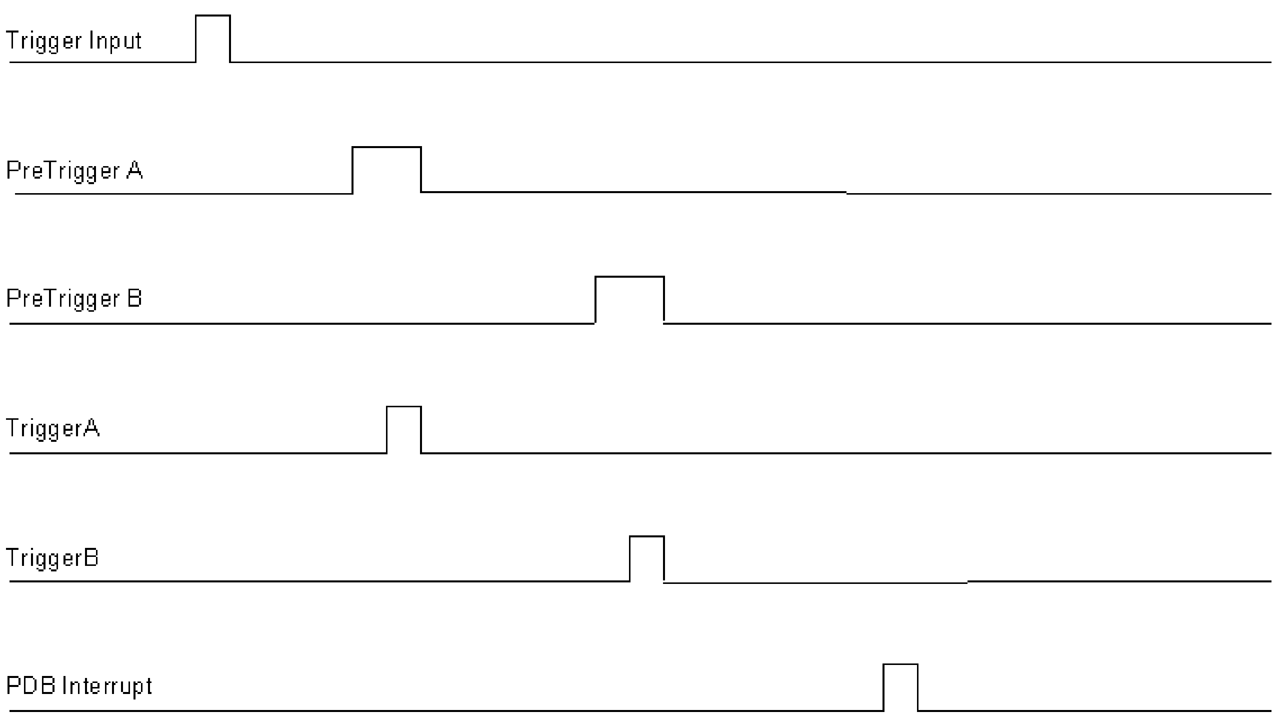

Figure 4. PDB trigger generation timing diagram

NOTE

Care must be taken with the timings. There is only one SAR within each ADC supporting only one conversion per ADC module at any time.

The time between PreTriggerA and PreTriggerB must be scheduled with a large safety margin for the previous conversion to be completed as shown in Equation 3. In this example a safety margin multiplier of 2.5 is used, but the system design needs to leave enough time only for the previous conversion to complete (accounting for any foreseeable error conditions).

$$CHnDELB – CHnDELA(TypicalConversionTime \times 2.5) / BusClock \qquad \textbf{Eqn. 3}$$

The same is true for the time between the 2nd PreTrigger and the PDB interrupt at time IDELAY as shown in Equation 4 with the same safety margin multiplier.

$$IDELAY – CHnDELB(TypicalConversionTime \times 2.5) / BusClock \qquad \textbf{Eqn. 4}$$

The PDB interrupt occurs if a comparison event has been detected; that is the 16-bit count equals the 16-bit IDELAY value when the module and PDB interrupts are enabled. IDELAY must be set for a safe time after the second conversion is complete. IDELAY can also be used as a single interrupt source to service all ADCs and to calculate the power, save data to memory, and so on.

The PDB module generates a sequence error interrupt (PDB_ERR) if a sequence error is detected on TriggerA or TriggerB. This happens if the CHnDELx has timed out before the previous ADC conversion has completed to enable the application to recover in the event of an error. This error is generated for any channel, cannot be disabled, and is non-maskable. It is imperative that an interrupt service routine be implemented for the PDB_ERR interrupt vector if the PDB is to be used for hardware triggering of the ADC.

4 ADC16 and Programmable Delay Block (PDB) Synchronization

Metering applications need to synchronize the time at which multiple ADC samples are taken with consideration to an external trigger or event. The intended function of the PDB on the MCF51EM256 is to provide controllable delays from either an external trigger or a programmable interval tick to the sample trigger input of one or more ADCs.

The MCF51EM256 has four independent ADCs to allow measurement of independent channels at the same time; Live1, Live2, Live3 (3-phase), and neutral. The ADC measures the line voltage, and then the line current a specified time later on each ADC module.

4.1 Scheduling Conversions

Figure 5 shows how the scheduling of conversions in a 3-phase electricity meter may be executed. The voltage conversions are all executed at the same time with PDB TriggerA and found in ADC Result Registers A. The current conversions are executed at a later time using TriggerB depending on the conversion time (plus guard band) of the ADC and the current transformers (CT) delay, which may be different for each phase, and the result found in ADC Result Register B. PDBCHnDLYA, and PDBCHnDLYB control the PreTriggers for the ADC channels.

The phase with the longest delay, in Figure 5 is Live2 and can use the COCOB flag in register ADC2SC1B to generate an interrupt. This interrupt can perform system operations, for example check that the results are valid, do power calculations, and store data in the secure memory. The PDB interrupt towards the end of the PDB count cycle is controlled by PDBIDLY to occur at a time relative to the counter starting, and can be used to prepare the PDB channels and the ADC modules for the next conversion.

Both interrupts are not necessarily required. The PDB interrupt has a higher priority than the ADC interrupts, so the scheduling may not include both interrupts.

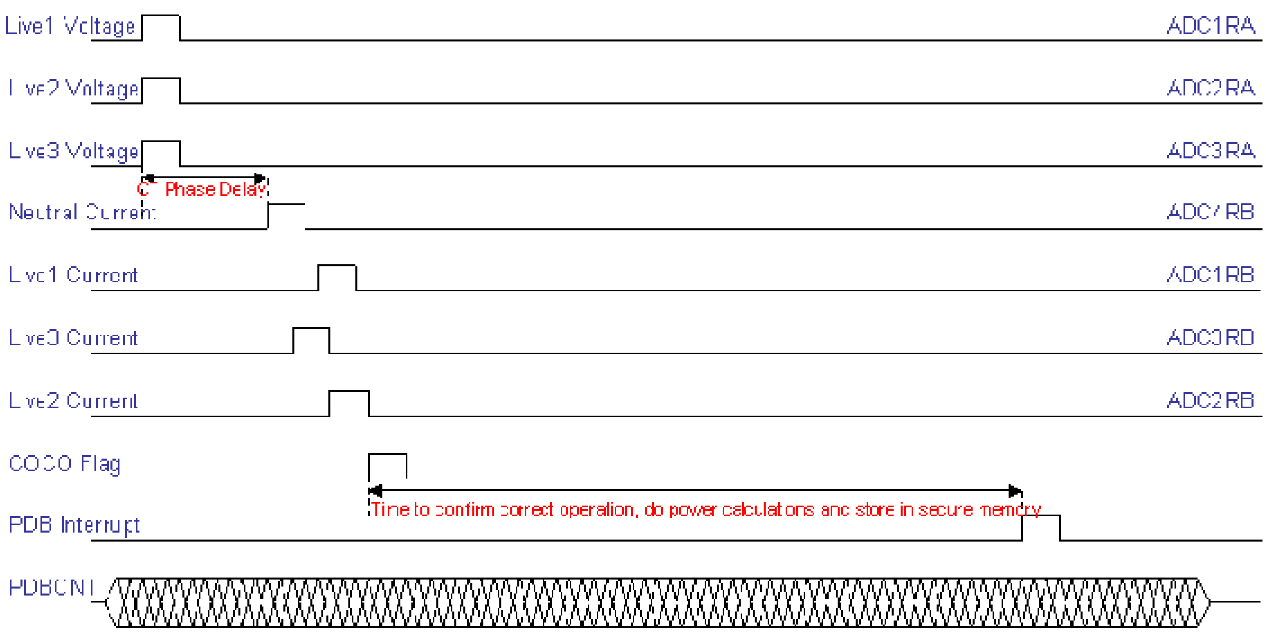

Figure 5. 3-phase electricity meter scheduling example timing diagram

The PDB modulus value (PDBMOD) depends on the electricity fundamental frequency in the country it is to be installed to enable the measurement of the 21st harmonic using the Nyquist theorem. In a 60 Hz country the sampling frequency can be 2 x 21 x 60 = 2.52 kHz. The maximum bus speed of the MCF51EM256 is 25 MHz. With a prescaler of 1, the PDB modulus value would be set as $26C0_{hex}$ = 25,000/2.52.

4.2 Synchronizing

AN3949SW contains the example software used to set-up a synchronized PDB and ADC for a system with synchronized inputs as described in this section.

4.2.1 Module Initialization

4.2.1.1 Analog-to-Digital Converter (ADC)

The worst case conversion time of the ADC for the settings must be known before the PDB trigger timings can be calculated.

NOTE

> This application note does not recommend ADC settings for best accuracy or timing. The application note titled *Differences Between Controller Continuum ADC Modules* (document AN3827) has information on how settings affect these parameters.

The majority of the ADC initialization would have been executed during the calibration of the module, but because the input channel, conversion mode, compare function values, resolution, and differential and single-ended settings have no effect on the calibration result, these may need to be set along with the trigger type select and interrupts, post-calibration.

The following calculation is based on these settings (20 MHz Bus) and taken from the accompanying software AN3949SW:

```
ADCCFG1 = (ADLPC_NORMAL|ADIV_2|ADLSMP_SHORT|MODE_16|ADICLK_BUS_2)
ADCCFG2 = (ADACKEN_DISABLED|ADHSC_HISPEED|ADLSTS_2)
ADCSC1A = (AIEN_OFF|DIFF_SINGLE|AD5)
ADCSC1B = (AIEN_OFF|DIFF_SINGLE|Bandgap)
ADCSC2 = (ADTRG_HW|ACFE_DISABLED|ACREN_DISABLED|REFSEL_EXT)
ADCSC3 = (ADCO_SINGLE|AVGE_DISABLED)
```

The conversion time for this configuration is 12.05 µs;

Calculated from BCT + High Speed Adder + Single Time Adder
= (25 ADACK) + (4 ADACK) + (5us + 5 ADACK + 5 bus clock cycles)
= (34 @ $5MHz_{ADACK}$) + (5 @ $20MHz_{BUS}$) + 5us

4.2.1.2 Programmable Delay Block (PDB)

With a 20 MHz bus to record the 21st harmonic of a 60 Hz signal, a modulus of $1F00_{hex}$ is needed (20 MHz and 2 x 21 x 60 Hz).

As shown in Figure 6, the first set of Triggers, A, occur at the start of the cycle. Figure 3 shows that the time between TriggerA and B in this example must be at least 2.5 times the conversion time; that is at least 30.125 µs (CHnDELB – CHnDELA (Typical Conversion Time x 2.5) / Bus Clock) which is 602.5 bus clock cycles with the ADC setting used in Section 4.2.1.1, "Analog-to-Digital Converter (ADC)," on page 12. The minimum TriggerB delay is the TriggerA delay + $25B_{hex}$.

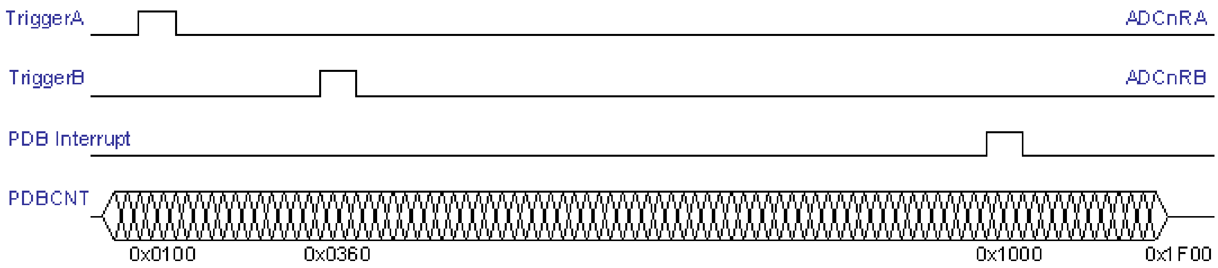

Figure 6. PDB timing

The interrupt delay (IDELAY) must be set to happen before the modulus match occurs, but early enough for the interrupt to be serviced along with all related activities.

In the example in the AN3949SW the PDB is set to run in continuous mode and counting is initiated by a single software-trigger. For a metering application example, refer to application note titled *MCF51EM256 Performance Assessment with Algorithms used in Metering Applications* (document AN3896).

4.2.1.3 Interpolation

Interpolation is a method used to construct unmeasured data points from a discrete set of measured data points. For example; in an electricity metering application, voltage can be assumed to be a perfect sine wave and therefore the error in the voltage using the interpolation method is insignificant (~ 0.0301%).

Figure 7 shows an example of a case where interpolation can be used. As shown in the **Standard** phase, if the voltage is measured at the beginning (V_{meas}), the current must be measured at I_{meas} (for example, a small phase delay due to the current transformers). The ADC conversion takes longer to complete than the allowed phase delay, then the current measurement needs to be taken before the first ADC channel is completed.

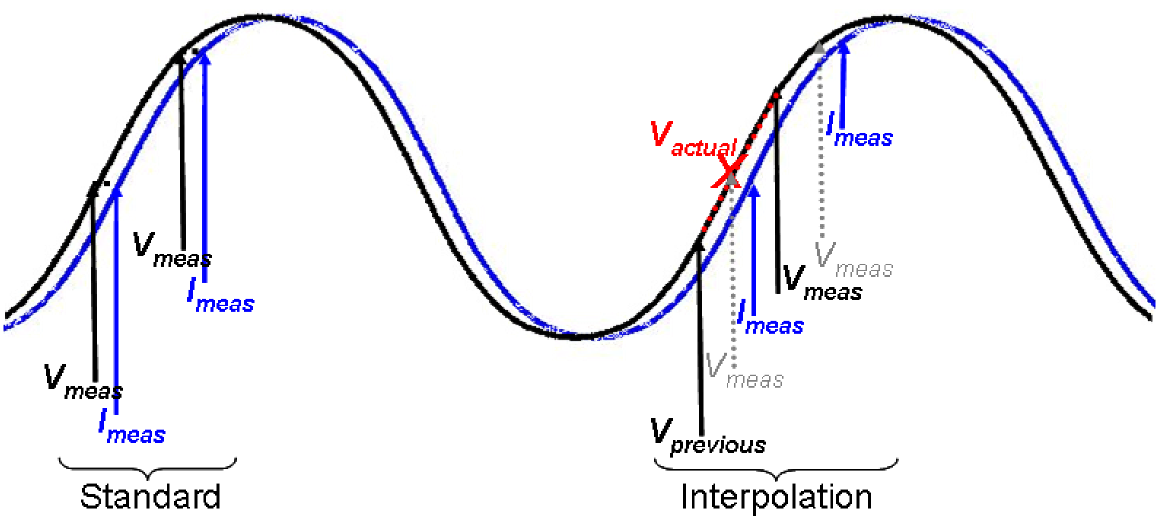

Figure 7. Interpolation example

In this case, when the phase delay for the current comes too close to the time for the voltage, move the voltage measurement to 50% of the measurement period (as shown in the **Interpolation** phase) and interpolate using the current and previous voltage measurements.

Below is a pseudo C example of how this can be achieved:

```
InterpolateMeasurements()
{
static Vmeas, Vprevious;
         Vmeas = readVoltageFromADC;
         Imeas = readCurrentFromADC;
         Vactual = (Vnow + Vprevious) >> 1;  // One shift right is divide by two
         Vprevious = Vmeas;                 // Save voltage reading for use next function call
}/ /returns Vactual and Imeas with correct phase shift
```

4.3 Monitoring Correct Operation

The PreTriggers pre-condition of the ADC to store the next conversion in either result register A or B. If a conversion was already running when another PreTrigger occurred in the same module, a corrupted sequence of results are recorded. Sequence errors, ERRA and ERRB are set when a PreTrigger is requested by the delay time-out before the last conversion is completed. This error generates a non-maskable interrupt.

The PDB sequence error interrupt must check to see what channel caused the error and then the application code can clear the flag and deal with the event accordingly. For example, save all valid data and restart the ADC and PDB, loosing only a few records or perform a soft reset.

NOTE

In a robust system environment this event is extremely rare and may be due to significant changes in ADC clocking frequencies. If this interrupt occurs frequently, the system integrity must be investigated. Clocks and external trigger events are most likely the cause.

Figure 8 shows the window where a second ADC trigger generates a sequence error event.

Figure 8. Sequence error window

5 Summary

Calibration is required to achieve the accuracies specified in the data sheet and to meet the systems accuracy requirements as shown in Figure 2. The calibration process uses some of the application's functional time and if not scheduled appropriately adds unnecessary overhead to the application. This application note has highlighted the best practice for calibration and explained how to minimize that latency. After the ADC is calibrated, the next item to address is the scheduling and synchronization of conversions using the ADC's hardware triggers from the PDB module. Figure 3, Figure 4, Figure 5 and Figure 6 illustrate the integration and application of the interlaced modules that enable system architects to generate robust synchronized conversions.

6 References

AN3896 — *MCF51EM256 Performance Assessment with Algorithms used in Metering Applications* by Paulo Knirsch

AN3827 — *Differences Between Controller Continuum ADC Modules* by Inga Harris

MCF51EM256 Reference Manual

MCF51EM256 Datasheet

AN3949SW — Referenced Software

How to Reach Us:

Home Page:
www.freescale.com

Web Support:
http://www.freescale.com/support

USA/Europe or Locations Not Listed:
Freescale Semiconductor, Inc.
Technical Information Center, EL516
2100 East Elliot Road
Tempe, Arizona 85284
+1-800-521-6274 or +1-480-768-2130
www.freescale.com/support

Europe, Middle East, and Africa:
Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
www.freescale.com/support

Japan:
Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku,
Tokyo 153-0064
Japan
0120 191014 or +81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:
Freescale Semiconductor China Ltd.
Exchange Building 23F
No. 118 Jianguo Road
Chaoyang District
Beijing 100022
China
+86 10 5879 8000
support.asia@freescale.com

For Literature Requests Only:
Freescale Semiconductor Literature Distribution Center
1-800-441-2447 or 303-675-2140
Fax: 303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com

Document Number: AN3949
Rev. 1
04/2010