Please note that Cypress is an Infineon Technologies Company.
The document following this cover page is marked as “Cypress” document as this is the company that originally developed the product. Please note that Infineon will continue to offer the product to new and existing customers as part of the Infineon product portfolio.

Continuity of document content
The fact that Infineon offers the following product as part of the Infineon product portfolio does not lead to any changes to this document. Future revisions will occur when appropriate, and any changes will be set out on the document history page.

Continuity of ordering part numbers
Infineon continues to support existing part numbers. Please continue to use the ordering part numbers listed in the datasheet for ordering.

THIS SPEC IS OBSOLETE

Spec No: 002-05434

Spec Title: AN205434 - FR, MB91470/480, A/D CONVERTER

Replaced By: NONE

AN205434

FR, MB91470/480, A/D Converter

Author:Cypress

This application note describes the functionality of the 8/10-bit and 12-bit Analog/Digital Converter (ADC) of the MB91470 and MB91480 Series and gives some examples. The 12-bit ADC allows high precision measurements of analog values in embedded applications.

Contents

1 Introduction

This application note describes the functionality of the 8/10-bit and 12-bit Analog/Digital Converter (ADC) of the MB91470 and MB91480 Series and gives some examples. The 12-bit ADC allows high precision measurements of analog values in embedded applications. The MB91470 provides two independent 12-bit ADC units (ADC3 + ADC4) with four channels each, as well as an 8/10-bit ADC with 12 channels. Most functions of the 10-bit ADC are equal to those of the 12-bit ADC. The MB91480 series provides three independent 8/10-bit ADC units (ADC0,1,2).

1.1 Key Features

- Minimum conversion time:
  12-bit: 2µs (@33MHz CLKP, 2.2us @40MHz CLKP, incl. sampling time)
  10-bit: 1.2µs (@33MHz CLKP, incl. sampling time)
- Dedicated result register for every ADC channel
- Status bits indicating missed or old samples for each channel
- Single-ended or differential input (12-bit ADC only)
- Interrupt or DMA transfer to memory after conversion selectable
- Triggered by software, external Pin (`ADTGx`), Multi-Function Timer or Reload Timer
- Additional activation by dedicated compare registers and a selectable Free-Run Timer / Multi-Function Timer (activate compare, compare clear or zero detect)

1.2 ADC units on MB91470/480 Series

The MB91470 series features two 12-bit AD converters with four channels each (units 3+4), as well as one 8/10-bit ADC with twelve channels (unit 2). The MB91480 series is equipped with three 8/10-bit ADC units (2x four channels + 1x eight channels)

1.3 Block Diagrams

The figure below shows the internal block diagram of the ADCs:

Figure 1. 8/10-bit ADC Block Diagram

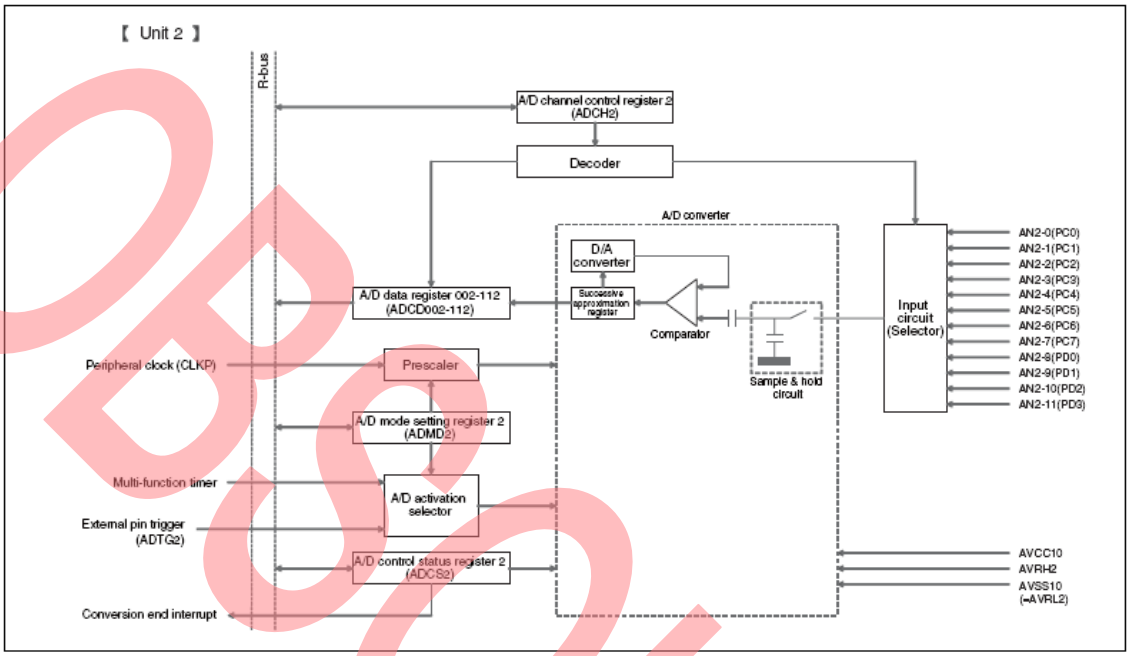
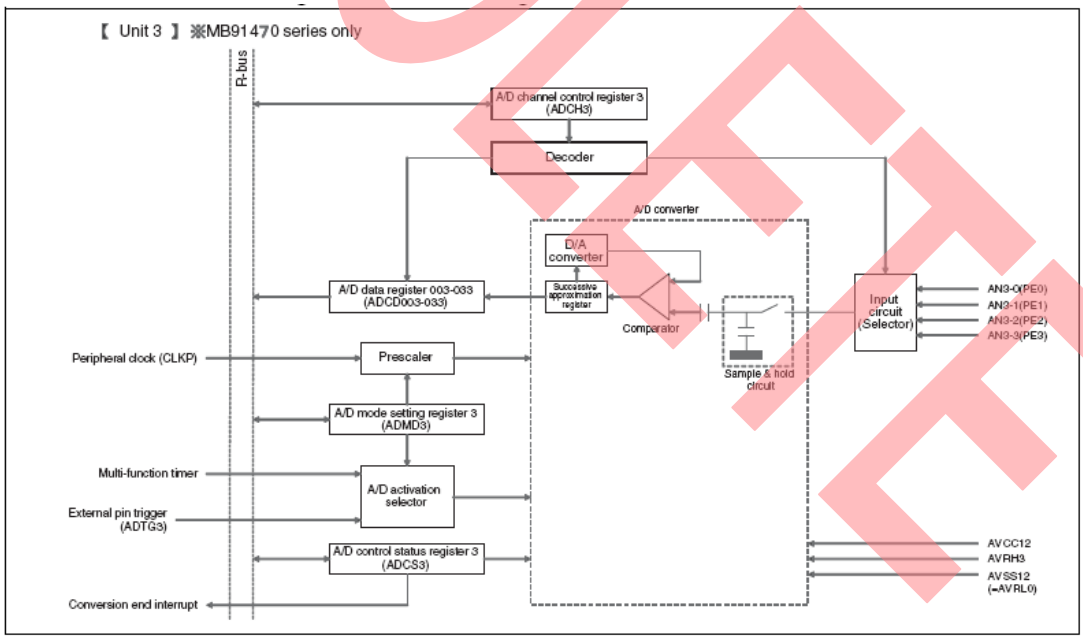


Figure 2. 12-bit ADC Block Diagram



2 Registers of the AD converter

Basic configuration of the ad converter. Unless otherwise noted (*), these register and configuration descriptions are applicable to the 8/10-bit ADC units 0-2 as well as the 12-bit ADC units 3+4.

2.1 Control Status Register (ADCSn)

This register controls the A/D converter and indicates its status.

Table 1. ADMDn

Bit No.	Name	Explanation	Initial Value	Value	Operation
15	BUSY	Busy Flag and Stop	0	0	Read: No A/D Conversion Write: Force Conversion Stop
				1	Read: A/D Conversion ongoing Write: No effect
14	INT	Interrupt Flag	0	0	Read: No A/D Data Write: Clear Flag
				1	Read: A/D Data and Interrupt Write: No effect
13	INTE	Interrupt enable	0	0	Interrupt disabled
				1	Interrupt, if A/D Data
12	PAUS	A/D Converter Pause	0	0	Read: No A/D Conversion Pause Write: Clear Bit
				1	Read: Pause occurred Write: No effect
11*	PINMD (12-bit)	Input pin mode select	0	0	Normal input mode
				1	Differential input mode
	S10 (8/10-b.)	AD resolution	0	0	10-bit resolution
				1	8-bit resolution
10	FuncSet	A/D function select bit	0	0	Separate result registers per channel
				1	One shared result register for all channels. Data protection function enabled.
9	STRT	Start Conversion	0	0	Always read; Write: no effect
				1	Start and Restart A/D Conversion
8	-	*Undefined*	X	0	*Reserved Bit , always write "0" to it*

When bit11 (PINMD) is set to '1' (MB91470 only), the 12-bit ADC is switched to differential input mode. In this mode, two ADC input channels are combined to one differential input as follows:

Pin name	Result register single-ended	Pin name differential	Result register differential
ANx-0	ADCD00x	ANx-0P	ADCD00x
ANx-1	ADCD01x	ANx-0N	
ANx-2	ADCD02x	ANx-1P	ADCD02x
ANx-3	ADCD03x	ANx-1N	

When using the differential input mode, the voltage on the neg. inputs must not exceed AVCC/2 or the voltage at the corresponding positive input.

2.2 A/D Mode Setting Register for 10-bit ADC (ADMD0,1,2)

Table 2. 12-bit ADC ADMDn register

Bit No.	Name	Explanation	Initial Value	Value	Operation
7, 6	MD1, 0	Operation Mode Select	0, 0	0,0	Single conv. mode 1; Reactivation during conversion is allowed
				0,1	Single conv. mode 2; Reactivation during conversion is not allowed
				1,0	Continuous Mode; Reactivation during Conversion not allowed
				1,1	Stop Mode; Reactivation during Conversion not allowed
5,4	STS1,STS0	Start trigger select	0	0,0	Software Start
				0,1	External pin (falling edge) or SW
				1,0	Multi-Function Timer or SW
				1,1	External pin, MFT or SW
3,2	CT1,CT0	Compare time	1,1	0,0	18 clk cycles (720ns @25MHz)
				0,1	24 clk cycles (720ns @33MHz)
				1,0	30 clk cycles (750ns @40MHz)
				1,1	60 clk cycles (1500ns @40MHz)
1,0	ST1,ST0	Sampling time	1,1	0,0	10 clk cycles (400ns @25MHz)
				0,1	13 clk cycles (390ns @33MHz)
				1,0	16 clk cycles (400ns @40MHz)
				1,1	32 clk cycles (800ns @40MHz)

The compare time has to be set to at least 720ns, while the sampling time must be greater or equal 390ns.

2.3 A/D Mode Setting Register for 12-bit ADC (ADMD3,4)

Table 3. 12-bit ADC ADMDn register

Bit No.	Name	Explanation	Initial Value	Value	Operation
7, 6	MD1, 0	Operation Mode Select	0, 0	0,0	Reserved
				0,1	Single Mode 2; Reactivation during Conversion not allowed
				1,0	Continuous Mode; Reactivation during Conversion not allowed
				1,1	Stop Mode; Reactivation during Conversion not allowed
5,4	STS1,STS0	Start trigger select	0	0,0	Software Start
				0,1	External pin (falling edge) or SW
				1,0	Multi-Function Timer or SW
				1,1	External pin, MFT or SW
3,2	CT1,CT0	Compare time	1,1	0,0	35 clk cycles (1.4µs @25MHz)
				0,1	42 clk cycles (1.27µs @33MHz)
				1,0	56 clk cycles (1.4µs @40MHz)
				1,1	112 clk cycles (2.8µs @40MHz)
1,0	ST1,ST0	Sampling time	1,1	0,0	20 clk cycles (800ns @25MHz)

				0,1	24 clk cycles (727ns @33MHz)
				1,0	32 clk cycles (800ns @40MHz)
				1,1	64 clk cycles (1600ns @40MHz)

The compare time has to be set to at least 1270ns, while the sampling time must be greater or equal 727ns.

2.4 ADC Data Registers (ADCD[ch][unit])

These registers store the digital value generated by the analog-to-digital conversion. This register contains the last converted value and is rewritten every time the conversion ends (depending on the ADCS:FuncSet bit, which selects single or multiple result registers).

Table 4. ADCDn

Bit No.	Name	Explanation	Initial Value	Operation
15	ERR	Error flag	1	This bit is set to '1' when reading it, and cleared when a new result is written to the corresponding ADCD register.
14	ERRST	Error status	0	If this bit is '0', it indicates that the same sample was read twice. If it is set ('1'), a sampled value was overwritten before the CPU had read it out.
13,12	-	Undefined Bits	X	Read value is not defined.
11 / 9* ... 0	D11/D9 ... D0	Data Bits	X	These bits contain A/D data after successful conversion (8/10-bit ADC: only D9…D0)

The MB91470/480 Series ADC (both the 8/10-bit as well as the 12-bit ADCs) provide dedicated result registers for every ADC channel. This functionality is controlled by the FuncSet bit it the ADMD register. If disabled, the ADC interrupt flag is set after every ADC conversion and the result can be treated by the ISR or moved by a DMA request. If the FuncSet bit is '0' (default), every ADC channel has a separate result register. The interrupt flag is set after all selected channels have been converted.

2.5 A/D Channel Control Register (ADCHn)

This register sets the start and end channel for conversion.

Table 5. ADCHn

Bit No.	Name	Explanation	Initial Value	Value	Operation
15-12	ANS3 ... ANS0	Start channel	0	0000	Channel 0
				0001	Channel 1
				...	...
				1011	Channel 11
11-8	ANE3 ... ANE0	End channel	0	0000	Channel 0
				0001	Channel 1
				...	...
				1011	Channel 11

The start channel always has to be smaller or equal the end channel (ANS <= ANE). To sample one channel only, set ANS = ANE.

He following channels can be selected:

MB91470 series: ADCH2 ANE/ANS = 0…11, ADCH3,4: ANE/ANS = 0…3

MB91480 series: ADCH0,1 ANE/ANS = 0…3, ADCH2 ANE/ANS = 0…9

If the 12-bit ADC is set to differential input mode (ADCS:PINMD = 1), only ANS1 and ANE1 are evaluated, because ch0+1 and ch2+3 are combined to one differential channel each.

2.6 Analog Input Control Register (AICR3,4)

The AICR register enables the analog inputs to the AD converter.

Table 6. AICR0,1,3,4

Bit No.	Name	Explanation	Initial Value	Value	Operation
7 … 4	–	*Reserved Bits*	X		Read value is undefined; writing has no effect
3 … 0	AN3E … AN0E	Analog input enable	1	0	Analog input disabled
				1	Analog input enabled

Table 7. AICR2

Bit No.	Name	Explanation	Initial Value	Value	Operation
15 … 12	–	*Reserved Bits*	X		Read value is undefined; writing has no effect
11 … 0	AN11E … AN0E	Analog input enable	1	0	Analog input disabled
				1	Analog input enabled

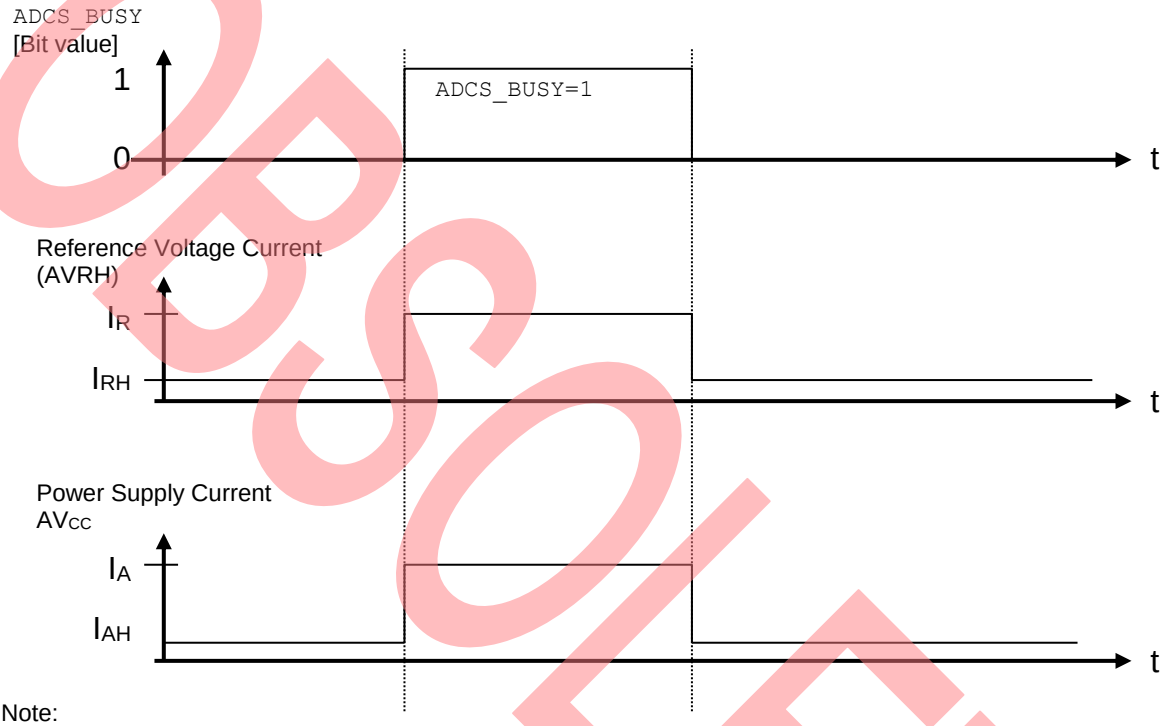
3 External circuitry for the ADC

Electrical power considerations

3.1 Power consumption

The power consumption (I_R, I_A) of the ADC increases in case a conversion is in progress (ADCS_BUSY = 1). While the ADC is halted (ADCS_BUSY = 0), only leakage current (I_{RH}, I_{AH}) flows. The following diagrams reflect this behaviour:

Figure 3. Power consumption and operating status of ADC



Note:

Please refer to the datasheet in order to get the absolute values of I_R, I_{RH}, I_A and I_{AH}.

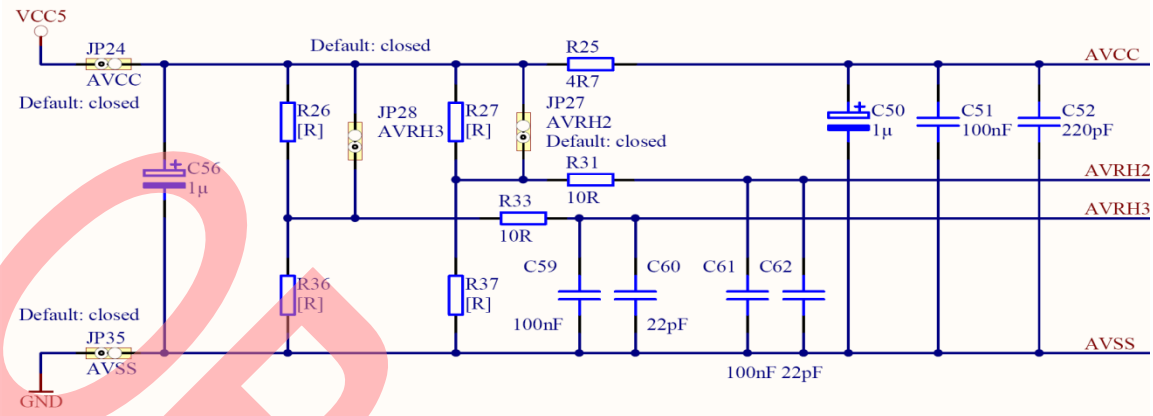
3.2 Power supply filtering

Due to the high resolution of the 12-bit ADC, the digital bit stream from the ADC output is sensitive to environment noise. For example, 1LSB corresponds to only about 1.2mV for U_{REF}=5V. Hence, the noise introduced from the external circuits must be considered and should be reduced as far as possible.

Any variations of the reference voltage AVRH will directly show on the ADC result. Therefore, it is important to keep the supply voltage to this pin constant and noise-free. Also the supply voltage of the AD converter is susceptible to noise and therefore usually should not be directly connected e.g. to the MCU's supply voltage. For many applications, it will be sufficient to use a simple RC low-pass filter for both the Vcc and AVRH voltage. The resistor value has to take the corresponding current into account and therefore will reside in the range of some Ohms. A good design practice is to use two capacitors in parallel for the RC filter, one bigger one filtering low frequency noise, and a smaller one filtering high frequency noise ((10nF–1µF)||(10pF-100pF)). Especially the small capacitor should be a ceramic type, since electrolyte capacitors often have a very poor performance at high frequencies.

The following schematic shows a possible filter network for both 12-bit ADCs of the MB91470 Series. It is used on the SK-91470-PMC1-MC.

Figure 4. A suggested connection for the power supplies (from SK91470-144PMC1-MC)

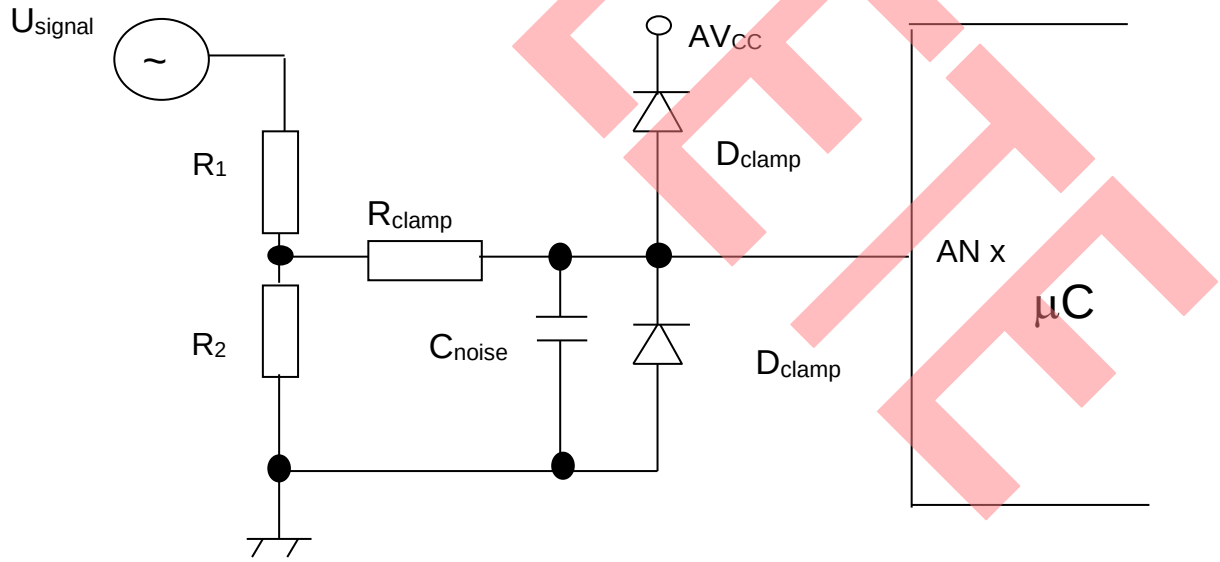


3.3 Input protection

As other CMOS inputs, also the analog inputs are sensitive to latch-up, which can occur if the applied voltage to an input pin exceeds the supply voltage, or different supply voltages are out of there respective range.

The minimum external circuit is a series ('clamping') resistor between signal source and input pin, which is also widely used for digital inputs to prevent possible voltage spikes or switching noise to cause latch-up. Since the internal sampling capacitor forms an RC low-pass filter with this clamping resistor, its value cannot be higher than some few kΩ without influencing the minimum sampling time. The internal ESD protection diodes of the MCU then can clamp the input voltage to AVCC and AVSS. However, the clamping current allowed for the MCU input is usually in the range of a few mA, so that additional precautions have to be taken in case a smaller clamping resistor is used. The following figure shows a typical circuit, which combines a voltage divider to adjust the source signal amplitude to the ADC range, an RC noise filter as well as two additional clamping diodes.

Figure 5. A typical external circuit for analog input



Additional information and considerations to ADC input circuits and sampling time considerations can be found in the application note *mcu-an-300215-e-v12-16fx_adc*, which can be found on the Fujitsu web page given at the end of this document. Even though it targets the ADC of the 16FX family, many aspects of ADC usage are applicable for ADCs in general.

4 ADC Software Examples

Examples for ADC usage

4.1 General ADC software example

```
/* THIS SAMPLE CODE IS PROVIDED AS IS AND IS SUBJECT TO ALTERATIONS. FUJITSU */
/* MICROELECTRONICS ACCEPTS NO RESPONSIBILITY OR LIABILITY FOR ANY ERRORS OR */
/* ELIGIBILITY FOR ANY PURPOSES.                                             */
/*                (C) Fujitsu Microelectronics Europe GmbH                   */
/*---------------------------------------------------------------------------*/

#define ADC3VAL0 (ADCD003 & 0x0fff)
#define ADC3VAL1 (ADCD013 & 0x0fff)
#define ADC3VAL2 (ADCD023 & 0x0fff)
#define ADC3VAL3 (ADCD033 & 0x0fff)

void init_adc3(void){ // initialize 12-bit AD converter unit 3

  ADCH3 = 0x03;       // Start Channel = 0, Stop Channel = 3
  ADMD3 = 0x8F;       // Continuous mode, maximal conversion time, sw trigger
  AICR3 = 0x0F;       // Enable AN3-0 to 3-3 inputs
ADCS3 = 0x82;       // No irq, seperate result registers, start conversion

}

void main (void) {

..
..
sprintf(outbuf,"\nADC3 values (12bit):  ch0 %d, ch1 %d, ch2: %d, ch3\
            %d\n\n",ADC3VAL0, ADC3VAL1, ADC3VAL2, ADC3VAL3);
uart_puts(uart,outbuf);
..
..
}
```

The above example demonstrates how to configure the ADC in continuous mode without interrupts. The maximum possible sampling and conversion time is used here. Because of the separate ADC result registers, the sampled data of all channels is directly available without setting up an ISR or DMA. The error bits of the AD result registers are not used in this example, and therefore are masked for ADC result register access.

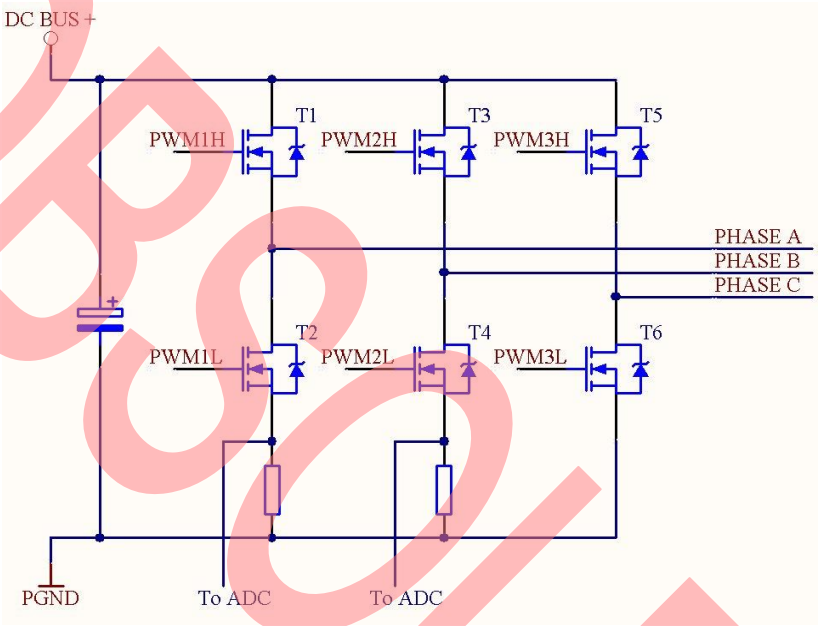
By using the separate result registers, it is easily possible to scan all ADC inputs sequentially within minimum time, and afterwards use the results. Therefore, multiple measurements with very small (and constant) delay in between are easily possible.

Another application for this function is a ‘quasi-differential’ ADC input: Two channels of the ADC are used together, and the results are subtracted from each other in software. This offers the benefit, that the result is available as signed integer in any number scaling, whereas the ‘native’ differential mode of the ADC does not support signed results and has some restrictions regarding input voltage range. Also it is possible to combine the result of more than two channels by any required function (multiplication, summation, possibly using the MAC unit / µDSP etc.) with the same small phase difference between the samples. Of course, since the input pins are not measured simultaneously but with a delay of one sampling time between each other, a phase error is introduced. Depending on the input frequency range it must be decided if this is acceptable for the application, and if it can be corrected by software. This phase error however also can be avoided by using two ADC units synchronously, e.g. triggered by the same Free-Running Timer as shown in the next example.

4.2 Synchronization to PWM generated by Multi-Function timer

Applications such as current control e.g. in a motor drive require ADC measurements that are precisely synchronized to PWM switching events, in order to minimize noise and to ensure a defined state of the inverter and thereby current path during the measurement. For example, if current is measured by shunt resistors placed between low-side power transistors and negative DC bus rail, the timing has to ensure that the low-side transistor is conducting during measurement. Often, two shunt resistors are used as shown below, with the third phase current being calculated from ∑i = 0. Therefore, two ADC samples have to be taken synchronously while the low-side transistors are active.

Figure 6. Phase current measurement (simplified)



As a part of the special motor control features of the MB91470 and MB91480 series MCUs, all ADCs can be triggered by several events of the Multi-Function Timer (MFT), which can generate various output waveforms, including complementary PWM. This is a necessary function for advanced motor control algorithms such as vector control, which relies on precise knowledge of the motor's phase currents. The ADC can be automatically triggered on compare clear and zero-detect events of a selectable Free-Running Timer (FRT). Additionally, separate, buffered activation compare registers allow to automatically trigger the ADC at arbitrary points during the PWM period, independently for the up- and down-counting phase of symmetric PWM if desired. This is especially useful for advanced techniques like single-shunt current reconstruction, which will be treated in a separate application note.

Depending on the PWM configuration (polarity, …), either the zero detect or the compare clear event will be the appropriate trigger for low-side current measurements (in the middle of two PWM pulses, when all low-side transistors are active). This implies that the duty cycle on the measured phases is not 100%, i.e. leaving a small 'window' for the low-side measurement. If this is not possible, one solution is to provide measurement hardware (shunt, etc) on all three phases, and selecting two for measurement according to the current duty cycles.

The following code snipped shows the initializations to be done to use a circuit similar to the one shown above. It only treats the ADC-relevant configurations.

```
/* THIS SAMPLE CODE IS PROVIDED AS IS AND IS SUBJECT TO ALTERATIONS. FUJITSU */
/* MICROELECTRONICS ACCEPTS NO RESPONSIBILITY OR LIABILITY FOR ANY ERRORS OR */
/* ELIGIBILITY FOR ANY PURPOSES.                                             */
/*                 (C) Fujitsu Microelectronics Europe GmbH                  */
/*---------------------------------------------------------------------------*/
#define ADC3VAL0 (ADCD003 & 0x0fff)
#define ADC4VAL0 (ADCD004 & 0x0fff)

void init_adc3(void){     // initialize 12-bit ADC unit 3 (current measurements)
  ADCH3 = 0x00;           // Start Channel = 0, Stop Channel = 0
  ADMD3 = 0x6A;           // Single mode, fastest conversion, trigger by MFT
  AICR3 = 0x01;           // Enable AN3-0 input
ADCS3 = 0xA0;           // Enable IRQ, use separate result registers
}

void init_adc4(void){     // initialize 12-bit ADC unit 4 (current measurements)
  ADCH4 = 0x00;           // Start Channel = 0, Stop Channel = 0
ADMD4 = 0x6A;           // Single mode, fastest conversion, trigger by MFT
  AICR4 = 0x01;           // Enable AN4-0 input
  ADCS4 = 0x80;           // No IRQ, separate result registers
}

void init_pwm(void){      // initialize OCU, FRT, WFG…
…

(remaining PWM initializations skipped here…)

ADTRGC0 = 0x66;         // trigger ADC3+4 on compare clear of FRT0
…
}

void __interrupt adc3_isr(void){  // also uses ADC4 Ch0 Value!
  ph_a_current = -(ADC3VAL0 - phase_current_offset_a) * CURRENT_SCALE;
                  // negative sign per definition: current flowing out of motor
                  // and through the shunt is negative
//  while (ADCS4_BUSY) __wait_nop();  // wait for ADC4 (usually not needed)
ph_b_current = -(ADC4VAL0 - phase_current_offset_b) * CURRENT_SCALE;
                // calculate phase c current (if needed):
 ph_c_current = -ph_a_current -ph_b_current;
 // some control algorithm…

 ADCS3 = 0xA0;    // clear interrupt flag and pause bit
                  // keep irq enable bit (Writing busy bit has no effect)
 }
```

In this code example, the result of both ADC units 3 and 4 are used in the ISR of ADC3. This can be done as long as all time-relevant configurations are identical, so that both ADCs start and stop synchronously, which usually will be the case.

5 Additional Information

Information about CYPRESS Microcontrollers can be found on the following Internet page:

http://www.cypress.com/cypress-microcontrollers

Related documents:

mcu-an-300215-e-v12-16fx_adc

The software example related to this application note is:

91470_adc_uart

It can be found on the following Internet page:

http://www.cypress.com/cypress-mcu-product-softwareexamples

Document History

Document Title: AN205434 - FR, MB91470/480, A/D Converter

Document Number:002-05434

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	-	WOFR	03/28/2008	CHa: First draft, partly based on 16FX ADC application note (by MWi)
			09/29/2008	First public version; combined 8/10-bit + 12-bit description, PWM sync example added
*A	5092203	WOFR	01/19/2016	Converted Spansion Application Note “MCU-AN-300108-E-V10” to Cypress format
*B	5560857	WOFR	12/20/2016	Obsoleted

Worldwide Sales and Design Support

Cypress maintains a worldwide network of offices, solution centers, manufacturer's representatives, and distributors. To find the office closest to you, visit us at Cypress Locations.

Products

Automotive	cypress.com/go/automotive
Clocks & Buffers	cypress.com/go/clocks
Interface	cypress.com/go/interface
Lighting & Power Control	cypress.com/go/powerpsoc
Memory	cypress.com/go/memory
PSoC	cypress.com/go/psoc
Touch Sensing	cypress.com/go/touch
USB Controllers	cypress.com/go/usb
Wireless/RF	cypress.com/go/wireless
Spansion Products	cypress.com/spansionproducts

PSoC® Solutions

psoc.cypress.com/solutions

PSoC 1 | PSoC 3 | PSoC 4 | PSoC 5LP

Cypress Developer Community

Community | Forums |Blogs | Video |Training

Technical Support

cypress.com/go/support

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709
Phone : 408-943-2600
Fax : 408-943-4730
Website : www.cypress.com