AN205553

F²MC-16FX Family, A/D Converter

This application note describes the functionality of the Analog/Digital Converter (ADC) and gives some examples.

Contents

1 Introduction....1
1.1 Key Features....1
2 The Analogue/Digital Converter....2
2.1 Block Diagram....2
2.2 Registers....3
3 Power supply of A/D converter....8
3.1 Power consumption....8
3.2 Noise consideration....9
4 Analogue input and related external circuits....10
4.1 External circuits for analogue input....10
4.2 Input Leakage current consideration....11
5 Sampling time consideration....13
6 Latch-up related to AV_{CC}/V_{CC} and large input signal....14
6.1 $AV_{CC} > V_{CC}$....14
6.2 $AV_{CC} < V_{CC}$....15
6.3 $U_{AIN} > AV_{CC}$ or V_{CC}....15
6.4 Conclusion....16
7 Input Impedance....16
7.1 Recharging and discharging the Sampling capacitor....16
8 ADC Example....18
8.1 ADC with interrupts....18
9 Additional Information....19
10 Document History....20

1 Introduction

This application note describes the functionality of the Analog/Digital Converter (ADC) and gives some examples.

1.1 Key Features

- Minimum conversion time per Channel: 1.7 µs at 20 MHz of CLKP1 (for $4.5V \le AV_{CC} \le 5.5V$)
- 8-bit or 10-bit conversion resolution
- Sequential channel conversion, once (Single Mode), continuous (Continuous Mode) and converts one channel, stops and waits for the next activation (Stop Mode)
- Interrupt generation after conversion selectable
- Interrupt can trigger DMA to transfer conversion result to memory
- Triggered by software, external Pin (ADTG) or Reload Timer 1

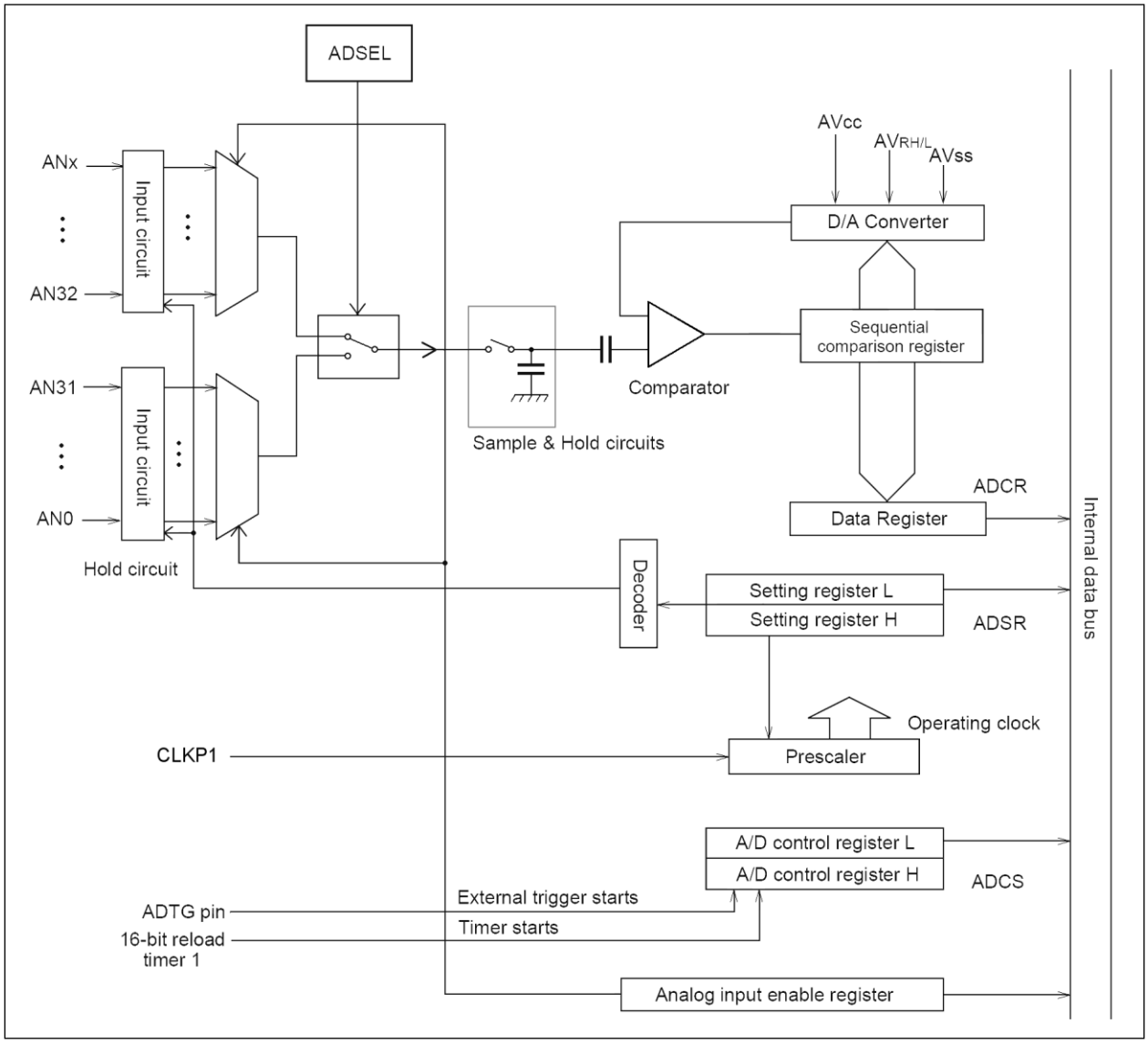
2 The Analogue/Digital Converter

The Basic Functionality of the Analogue/Digital Converter

2.1 Block Diagram

Figure 1 shows the internal block diagram of the ADC.

Figure 1. ADC Block Diagram



x is last/highest ADC channel available on the particular device.

2.2 Registers

2.2.1 Control Status Register (ADCS)

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

Table 1. ADCS

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, 10	STS1,0	Start Source Select	0, 0	0, 0	ADC Activation by Software
				0, 1	ADC Activation by Software and ADTG Pin
				1, 0	ADC Activation by Software and Timer
				1, 1	ADC Activation by Software, ADTG Pin, and Timer
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*
7, 6	MD1, 0	Operation Mode Select	0, 0	0, 0	Single Mode 1; Reactivation during Conversion allowed
				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	S10	10-Bit Mode	0	0	10-Bit Conversion Mode
				1	8-Bit Conversion Mode
4 … 1	-	*Undefined*	X	0	*Reserved Bit , always write "0" to it*
0	-	*Reserved*	0	0	*Reserved Bit , always write "0" to it*

* These bits return "1" during Read-Modify-Write instruction.

2.2.2 Date Register (ADCR)

This register stores digital value generated as a result of conversion. This register contains the last converted value and is rewritten every time the conversion ends.

Table 2. ADCR

Bit No.	Name	Explanation	Value	Operation
15 … 10	–	-	0	These bits always return “0”
9 … 0	D9 … D0	Data Bits	–	These bits contain A/D data after successful conversion

2.2.3 Setting Register (ADSR)

This registers sets the sampling and conversion time, also sets the starting and ending channel for conversion.

Table 3. ADSR

Bit No.	Name	Explanation	Initial Value	Value	Operation
15 … 13	ST2 … ST0	Sampling Time Setting	0, 0, 0	0, 0, 0	4 peripheral clock cycles
				0, 0, 1	6 peripheral clock cycles
				0, 1, 0	8 peripheral clock cycles
				0, 1, 1	12 peripheral clock cycles
				1, 0, 0	24 peripheral clock cycles
				1, 0, 1	36 peripheral clock cycles
				1, 1, 0	48 peripheral clock cycles
				1, 1, 1	128 peripheral clock cycles
12 … 10	CT2 … CT0	Compare Time Setting	0, 0, 0	0, 0, 0	22 peripheral clock cycles
				0, 0, 1	33 peripheral clock cycles
				0, 1, 0	44 peripheral clock cycles
				0, 1, 1	66 peripheral clock cycles
				1, 0, 0	88 peripheral clock cycles
				1, 0, 1	132 peripheral clock cycles
				1, 1, 0	176 peripheral clock cycles
				1, 1, 1	264 peripheral clock cycles
9 … 5	ANS4 … ANS0	Starting Channel Setting	0, 0, 0	0, 0, 0	AN0 (ADECR_ADSEL = 0) AN32 (ADECR_ADSEL = 1)
				0, 0, 1	AN1 (ADECR_ADSEL = 0) AN33 (ADECR_ADSEL = 1)
				0, 1, 0	AN2 (ADECR_ADSEL = 0) AN34 (ADECR_ADSEL = 1)
				. . .	. . .
				1, 1, 1	AN31 (ADECR_ADSEL = 0)
4 … 0	ANE4 … ANE0	Ending Channel Setting	0, 0, 0	0, 0, 0	AN0 (ADECR_ADSEL = 0) AN32 (ADECR_ADSEL = 1)
				0, 0, 1	AN1 (ADECR_ADSEL = 0) AN33 (ADECR_ADSEL = 1)
				0, 1, 0	AN2 (ADECR_ADSEL = 0) AN34 (ADECR_ADSEL = 1)
				. . .	. . .
				1, 1, 1	AN31 (ADECR_ADSEL = 0)

Note: Always use word access to write to this register.

Please refer the datasheet for the highest ADC channel number available on the device.

Sampling/Conversion Time

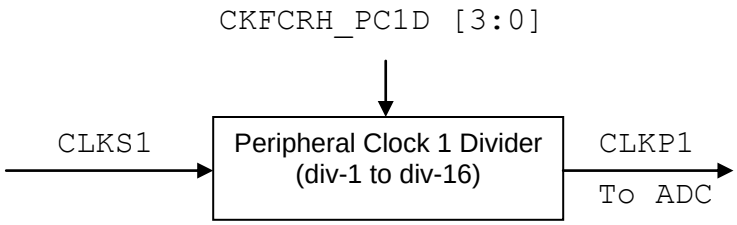
The sampling time must be equal to or greater than 0.5 µs. The Conversion time must be at least 1 µs. Therefore an overall A/D conversion time could be 1.5 µs, but this depends on the used peripheral clock.

Assume a peripheral clock of 16 MHz (62.5 ns cycle time). The sample cycle number then has at least to be 8 (0.5 µs) and the conversion cycle number can be 22 (1.375 µs). Therefore the overall A/D conversion time is 1.875 µs.

All these settings are valid for 4.5 V ≤ AV_{CC} ≤ 5.5 V. Please see the hardware manual for external impedance considerations.

Please consider that the sampling/conversion time settings also depend on the settings in the peripheral Clock 1 divider.

Figure 2. ADC Clock



Start/Stop Channels

- Use start channel = end channel for conversion of only one channel.
- If start channel is greater than end channel, conversion starts form start channel to the highest available channel, then from channel 0 to the end channel. For products with less than 32 A/D converter channels, the conversion results can be incorrect. To prevent this, do not set ANS larger than ANE.
- It is not possible to have a start channel < 32 and an end channel ≥ 32. If this is needed, please use the following sequence:
  - Set `ADECR:ADSEL` = 0 to convert channels from required start channel to channel 31.
  - Set `ADECR:ADSEL` = 1 and convert channel 32 to the required end channel.

2.2.4 ADC Extended Configuration Register (ADECR)

If there are more than 32 ADC channels available, the ADECR switches between upper and lower ADC channel set. The deselected set cannot be accessed then. It also selects A/D converter's high and low reference voltage, if available.

Table 4. ADECR

Bit No.	Name	Explanation	Initial Value	Value	Operation	
7 … 3	–	*Reserved Bits*	X	0	Read value is undefined; Write "0" to these Bits.	
2, 1	LSEL,HSEL	Reference Voltage	0,0		Low reference voltage	High reference voltage
				0,0	AVRL/AVRH2	AVRH
				0,1	AV_{SS}	AVRH
				1,0	AV_{SS}	AVRH
				1,1	AV_{SS}	AVRL/AVRH2
0	ADSEL	ADC Channel Set Select	0	0	Select ADC Channels 0 - 31	
				1	Select ADC Channels ≥ 32	

It should be noted that the reference voltage switch is available on selected devices. In the devices where the reference voltage switch is unavailable, bits LSEL and HSEL are unavailable and high reference voltage pin is AVRH and low voltage reference pin is AVSS always.

2.2.5 Analog Input Enable Register (ADER0 – ADERn)

The ADER0 - ADERn selects whether a corresponding pin is used as an analog input or a digital I/O channel. The following formula shows the relation between total number of ADC channels and the highest ADER register:

$$n = \frac{x}{8} - 1$$

Where, x is the total number of ADC channels available on the device

n is the highest ADER register

The following table shows ADER0.

Table 5. ADER0

Bit No.	Name	Explanation	Initial Value	Value	Operation
7 … 0	ADE7 … ADE0	ADC Input Selection	0	0	Digital I/O Port enabled
				1	Analog Input enabled

The following table shows ADER1.

Table 6. ADER1

Bit No.	Name	Explanation	Initial Value	Value	Operation
7 … 0	ADE15 … ADE8	ADC Input Selection	0	0	Digital I/O Port enabled
				1	Analog Input enabled

The following table shows ADERn.

Table 7. ADERx

Bit No.	Name	Explanation	Initial Value	Value	Operation
7 … 0	ADE(X-1) … ADE(X-8)	ADC Input Selection	0	0	Digital I/O Port enabled
				1	Analog Input enabled

Hence if total number of ADC channels available on a particular device is 24 then the highest ADER register is ADER2 and bits of the same would be ADE16 to ADE23.

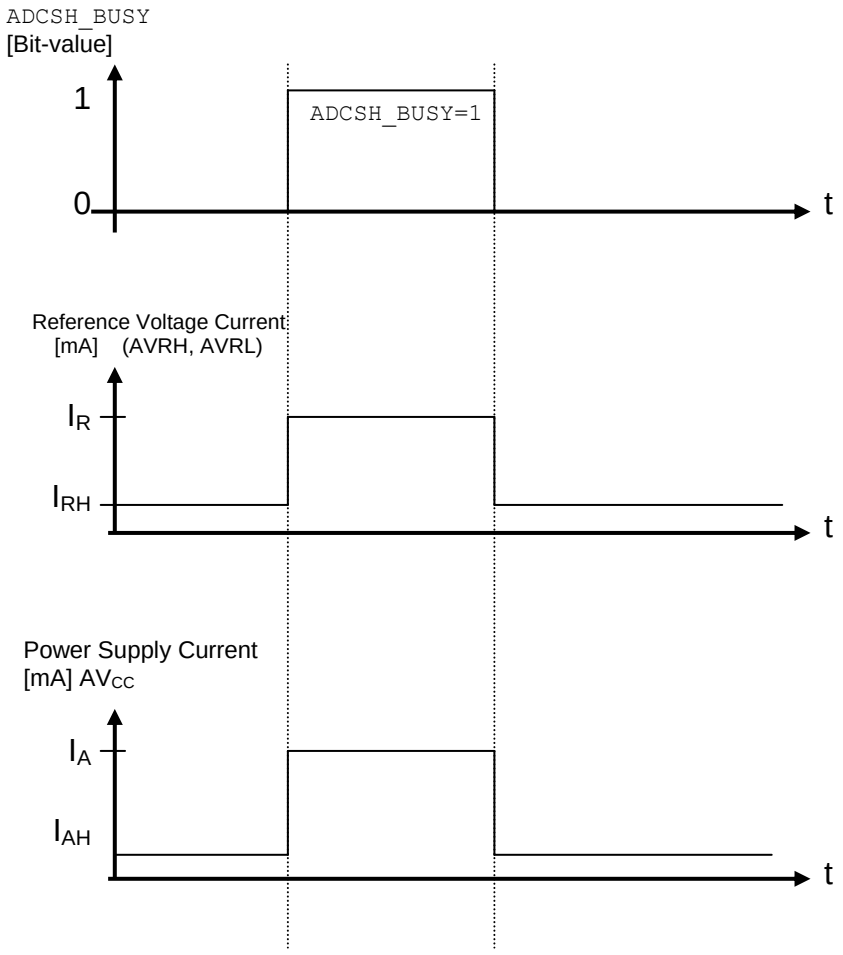
3 Power supply of A/D converter

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 (ADCSH_BUSY = 1). While the ADC is halted (ADCSH_BUSY = 0), only leakage current (I_{RH}, I_{AH}) flows. The following diagrams reflect this behavior:

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 Noise consideration

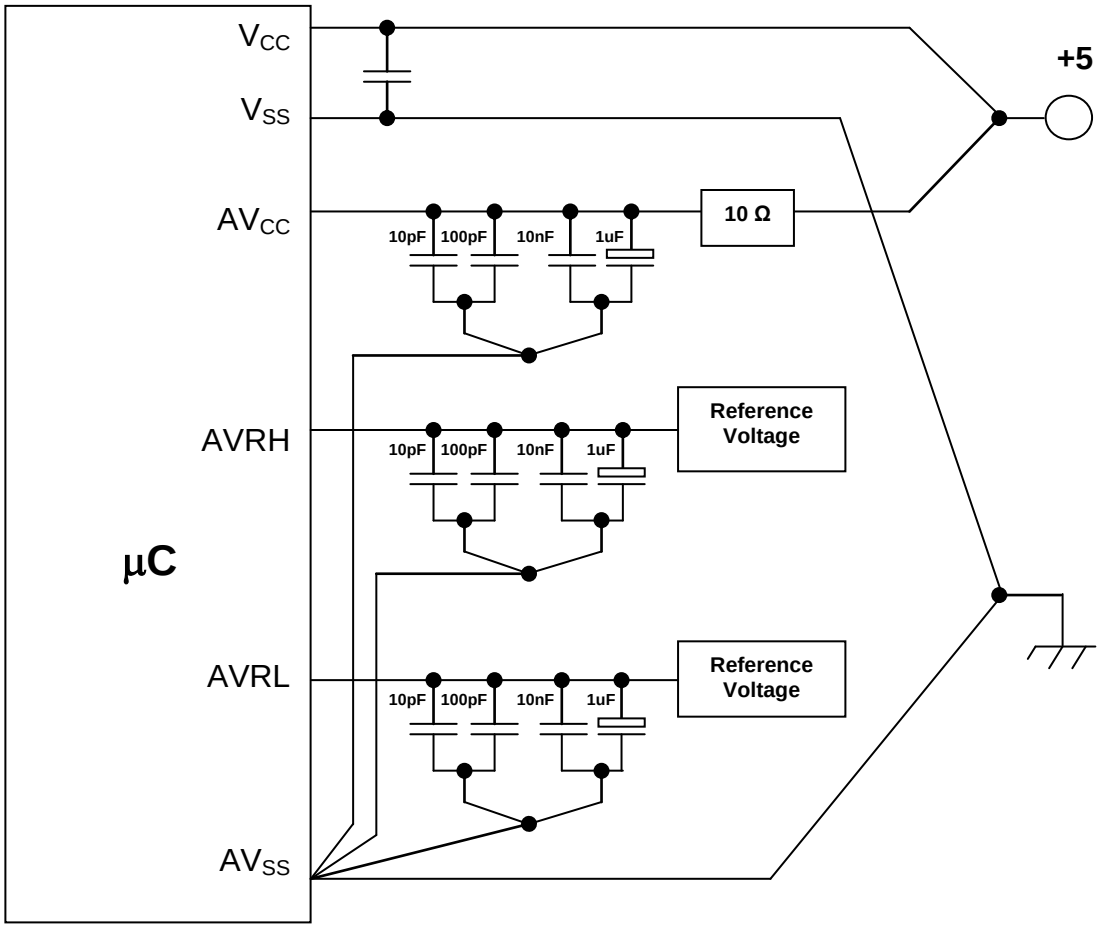
Cypress microcontroller has implemented an embedded 10-bit Successive Approximated Register (SAR) ADC. Due to the high resolution, the digital bit stream from the ADC output is sensitive to the environment noise. For example, 1LSB corresponds to only 4.9mV for UREF=5V. Hence, the noise introduced from the external circuits must be considered and should be reduced to the minimum as possible.

The reference voltage U_{REF}, which is equal to AVRH-AVRL (depends on `ADECR` setting if reference voltage switch is available), is connected to the weighted capacitor array and the resistor array of the ADC. The noise coupled to AVR will not be rejected by ADC. This noise will be added to the U_{REF} directly, introducing an error with a ratio of U_{Noise}/U_{REF}. For example, to keep the error caused by this kind of noise below 0.1LSB, the noise level of U_{REF} must be kept within 0.49mV.

As a result, the pin AVRH and AVRL pins must be connected with low impedance. In practice often a simple low-pass RC-filter is used for noise reduction. In this case the reference voltage supply current (see datasheet) has to be taken into account when calculating the resistor of the filter, in order to minimize the voltage drop while converting. Normally two capacitors in parallel are recommended, one filtering low frequency noise, the other one filtering high frequency noise ((10nF–1µF)||(10pF-100pF)). In most cases, this configuration suppresses the noise efficiently. If very high frequency noise appears in the environment, an additional noise filter such as a dedicated π mode RC filter might be useful.

The analogue power path AV_{CC} supplies the internal voltage comparator and the analogue switches of the ADC, while the V_{CC} path supplies all the digital parts in the microcontroller. Internal parasitic capacitors may couple noise from AV_{CC} to the internal voltage comparator of the ADC. For this reason, also AV_{CC} should not be connected directly to V_{CC} but filter should be used, too. For more efficient noise filtering the same configuration as for AVRH is recommended.

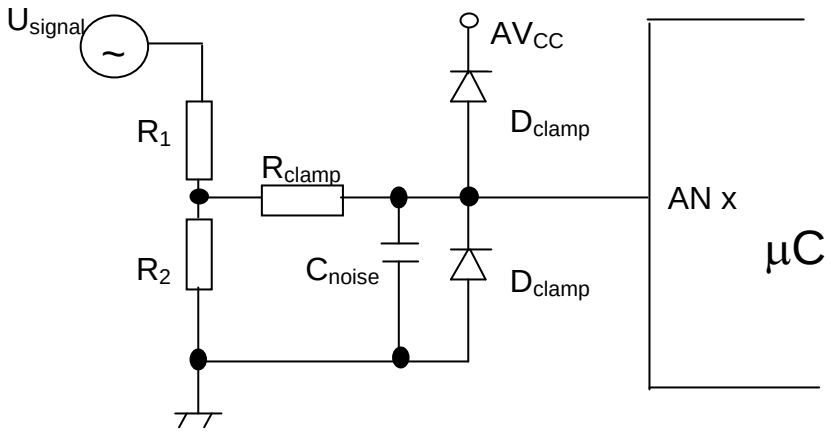
Figure 4. Suggested connection for the power supplies



4 Analogue input and related external circuits

4.1 External circuits for analogue input

Figure 5. A typical external circuit for analogue input



To protect the analogue pins to suffer from an over-voltage, the so-called “clamping resistor” is usually added to the input pins. The minimum value of the resistor can be chosen as

$R_{clamp} = U_{overvoltage}/I_{clamp}$,

Where, I_{clamp} is the specified maximum clamp current in the data sheet.

For some applications, a large clamp resistor is sometimes unacceptable. As a compromise, an external clamping diode with low leakage current could be added between the input pin and AV_{CC} pin.

In some cases, the sensor has been biased with a voltage supply higher than the maximum allowed voltage for the microcontroller. For example, in the automotive applications, the sensors could be biased directly with the car battery, which exhibits a voltage of 12V/24V. A resistor divider consisting of R_1/R_2 is commonly used to tail the sensor voltage signal “seen” on the pin down to the value which is equal or smaller than AV_{CC}/V_{CC} (see Figure 5).

The ratio between R_1 and R_2 should satisfy the following constrain:

$$\frac{R_1}{R_2} \geq \frac{U_{Signal}}{AV_{CC}} - 1$$

Other factor which influences the size dimension of R_1, R_2 and R_{clamp}, is related to current consumption budget and the input signal noise suppressing. The second factor will be discussed here with more detail. The signal from the sensors could be also noisy. The noise, which has a time constant smaller than the sampling time $T_{sampling}$, is transparent to the ADC, resulting distorted output. In this case, an additional dedicated bypass capacitor together with the clamping resistor or resistor divider, works as a low pass filter. A larger capacitor will lower the AC impedance and will be more effective at shunt away the noise signal. Generally, the time constant of this low pass filter (R_{clamp} + R_1 || R_2) x C_{noise} should be chosen considerably larger than the sampling time (5 to 10 times larger with a rule of thumb).

However, at the same time this time constant should be also considerably smaller than the one of the sensor signal, depending on the applications. In this way, the analogue pin is able to follow the dynamic changes, which the ADC is being used to track. These, along with the dimension of R_1/R_2 or R_{clamp} must be considered when choosing the capacitor dimension to avoid rolling off any high frequency signal components of interest.

4.2 Input Leakage current consideration

The analogue input pins show a small leakage current, whose maximum value is about 3μA and ranged from 3μA down to 1μA depending on the temperature. The leakage current, which flows through the external resistor, introduces an undesired voltage drop. This error voltage is a function of the external resistor and the leakage current itself. The following example shows a dimension of the resistor with this factor taken into consideration. For the case of using a resistor divider to reduce the error due to leakage current, the size of $R_1 \parallel R_2 + R_{clamp}$ should not be chosen too large and should be according to the following equation:

$$R_1 \parallel R_2 + R_{clamp} \leq \frac{U_{LSB}}{I_{leakage}}$$

Note:
$U_{LSB} = U_{REF} / 1024$

To keep the error smaller than one LSB for a leakage of 3μA, the size of $R_1 \parallel R_2 + R_{clamp}$ should be smaller than 1.6kΩ. As the leakage current drops down to 1 μA, the value of R1 || R2 + R_{clamp} can be chosen as large as 5kΩ. This is considering U_{REF} of 5V.

It is found in the test that the leakage current consists of two parts: one is due to the leakage current of the input ESD structure. Another leakage current appears only as the multiplexer is switched on during the sampling time, whose contribution is usually considerably larger than the one created from ESD structure. The second leakage current can be regarded as a noise during the sampling time by the bypass capacitor, which is commonly used to filter the noise from the sensor input. If this capacitor is large enough, it can absorb most of the second leakage current during the sampling time, eliminating its contribution to the error voltage.

Figure 6. Leakage current flowing to the analogue input pin

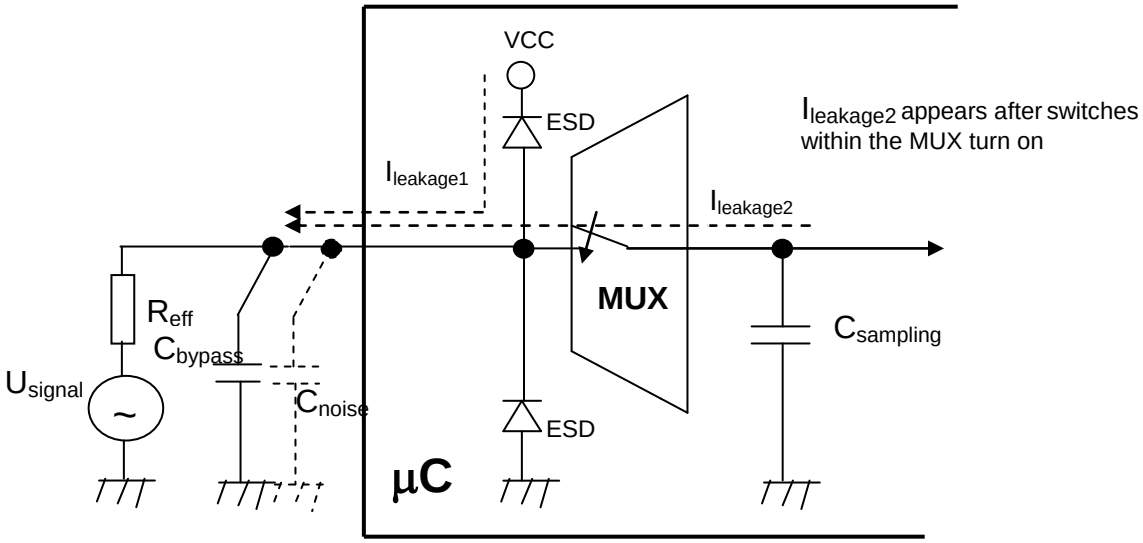
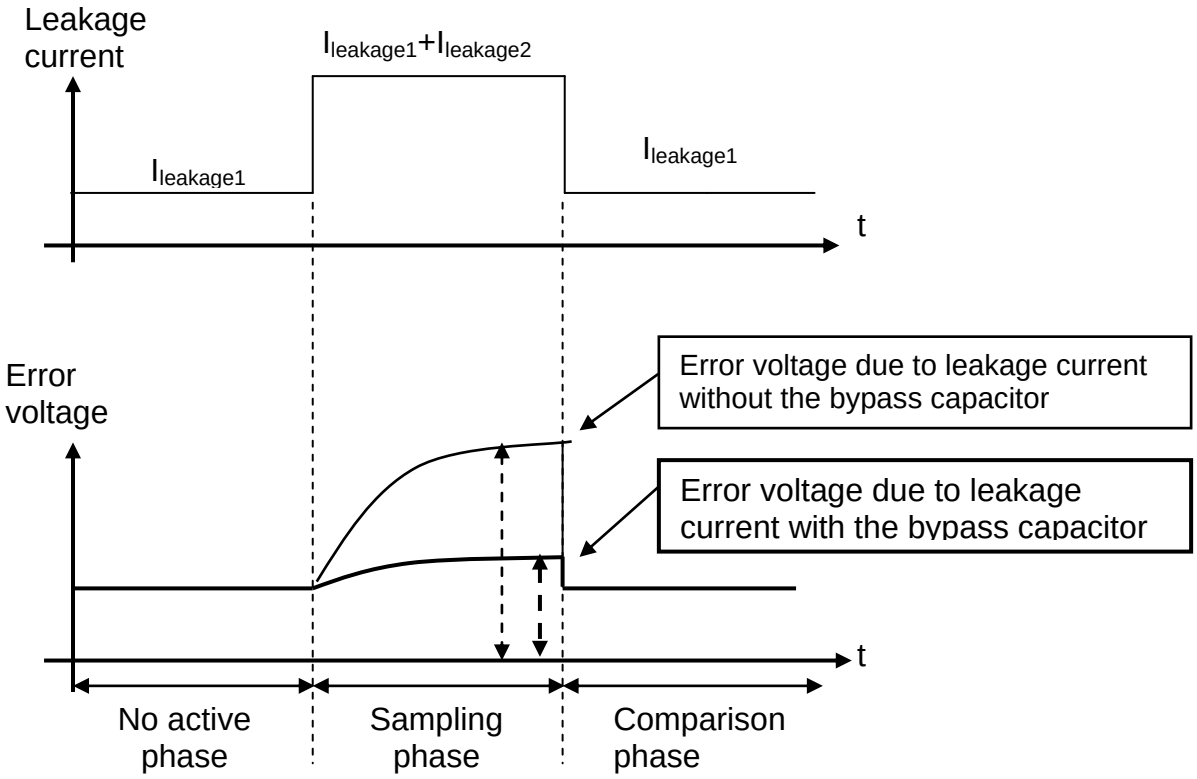


Figure 7. Reducing the leakage current with the bypass capacitor



To show the effect of the bypass capacitor on reducing the leakage current error, we take a sampling time of 5μs and a leakage current of 3μA as an example. If we want to keep the voltage drop due to the second leakage current small than 0.5 LSB, the minimum size of the bypass capacitor should be chosen as:

$$C = \frac{3\mu A \times 5\mu s}{4.9mV/2} \approx 6nF$$

5 Sampling time consideration

Sampling time consideration

Cypress applies an embedded 10-bit successive approximation register ADC with an internal integrated sampling and hold stage. The signal will charge the sampling capacitor at first and then the voltage signal on the sampling capacitor will be evaluated by the 10-bit ADC successively. The time to charge the sampling capacitor to its final value equal to the signal level is a function of the sampling capacitor $C_{sampling}$, the external resistor and the internal switch on-resistor.

To reduce the error caused by the limited sampling time to an acceptable level, the sampling time should be chosen much larger than the time constant to charge the sampling capacitor. For example, if we choose a sampling time with a factor 7 of the RC constant, namely, $7\times(R_{extern}+R_{switch})\times C_{sampling}$. Then the error amounts to $e^{-7}\times U_{REF}$, corresponding $0.94\times U_{LSB}$ only.

For MB96340 Series, the on-resistor of the transmission gate amounts to 2.25kΩ and the sampling capacitor $C_{sampling}$ equals to 10.7pF at AV_{CC} = 5V. For an external resistor of 2.25kΩ, the sampling time should be chosen larger than $7\times(2.25k\Omega+2.25k\Omega)\times 10.7\,pF = 0.38\mu s$ by using above thumb rule.

Figure 8. Block diagram for ADC in MB96340 Series

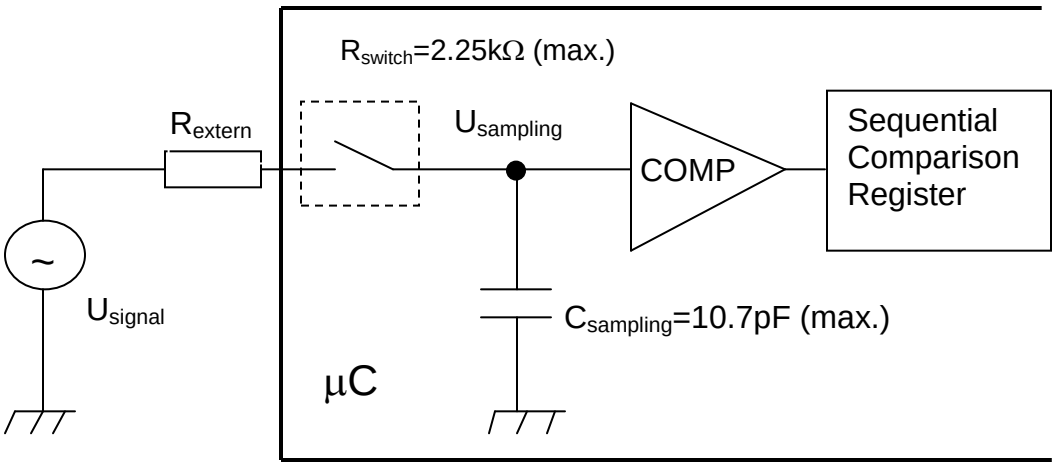
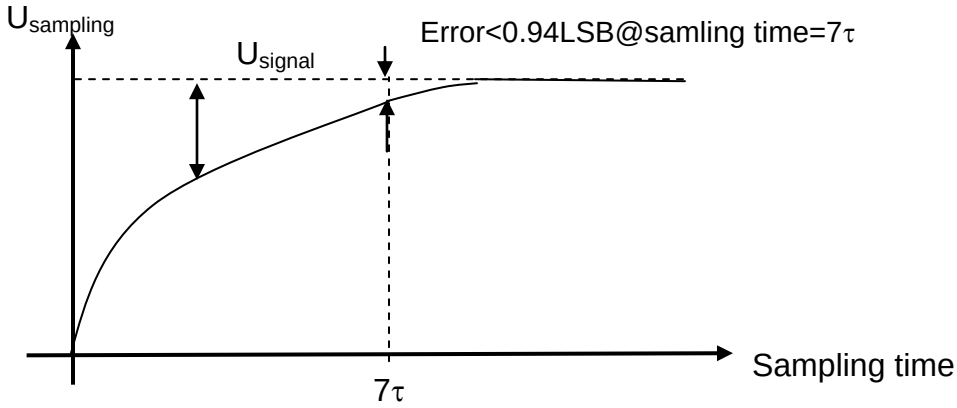


Figure 9. Error related to sampling time



For the Cypress microcontroller the sampling time can be set by defining the register bit ST2 to ST0 of the ADC data register (ADCRH). Please refer to the hardware manual.

Please also refer Chapter 7: Input Impedance

6 Latch-up related to AV_{CC}/V_{CC} and large input signal

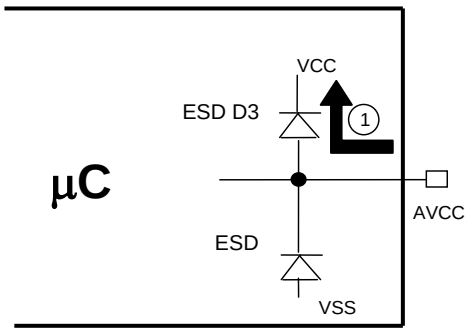
Recommendations to prevent Latch-Up

Latch-up conditions can permanently damage the device and must be avoided. It is up to the application to assign any precautions in order to avoid any latch-up condition.

6.1 $AV_{CC} > V_{CC}$

Figure 10. Latch-up in case $AV_{CC} > V_{CC}$

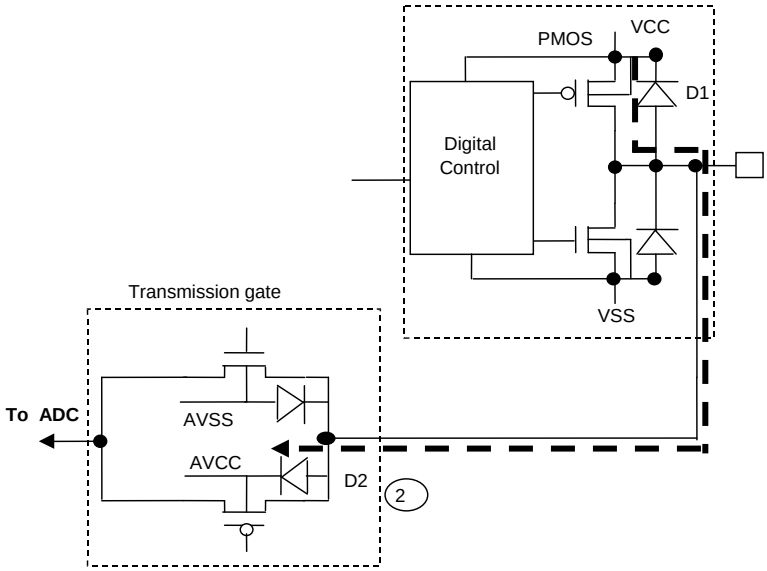
`



Latch-up can happen if AV_{CC} becomes larger than V_{CC}. This might be related to the application cases that V_{CC} is switched on later than AV_{CC} or V_{CC} is switched off earlier than AV_{CC}. The ESD diode D3 becomes forward biased, introducing a possible latch-up.

6.2 $AV_{CC} < V_{CC}$

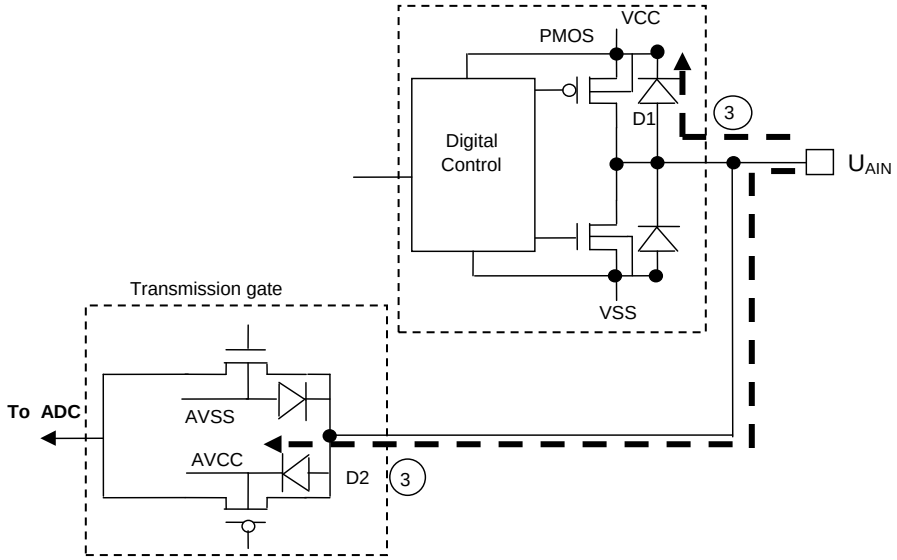
Figure 11. Latch-up in case $AV_{CC} < V_{CC}$



This case happens if the analogue pin is used as a digital output pin, the output level is “H”, and at the same time, AVCC is switched off. In this case, the PMOS in the output is on and the parasitic diode D2 of the transmission gate within the analogue multiplex becomes forward biased. A quiescent current flows through PMOS and D2. In case that a latch-up does not happen, a reliable logic “1” should not be expected at the output, due to the load diode D2.

6.3 $U_{AIN} > AV_{CC}$ or V_{CC}

Figure 12. Problem in case UAIN > AV_{CC} or V_{CC}



If U_{AIN} becomes larger than V_{CC} or AV_{CC}, then the ESD diode D1 or D2 will be forward biased. A latch-up can happen. Even if a latch-up does not always happen in this case, the input signal, which exceeds V_{CC} or AV_{CC}, cannot be converted by the ADC properly.

6.4 Conclusion

It is strongly suggested that AV_{CC} and V_{CC} should be DC short circuit together to avoid any possible latch-up.

With the presence of AV_{CC} and V_{CC} voltage, an analogue input signal, which is smaller than AV_{CC} and V_{CC} can be always put on the analogue pins, independent on the MCU modes.

Latch-up conditions can permanently damage the device if the related specified currents are exceeded. So Latch-up conditions must be avoided under all circumstances. It is up to the application to assign any precautions in order to avoid any latch-up condition.

7 Input Impedance

ADC behavior on High Input Impedances

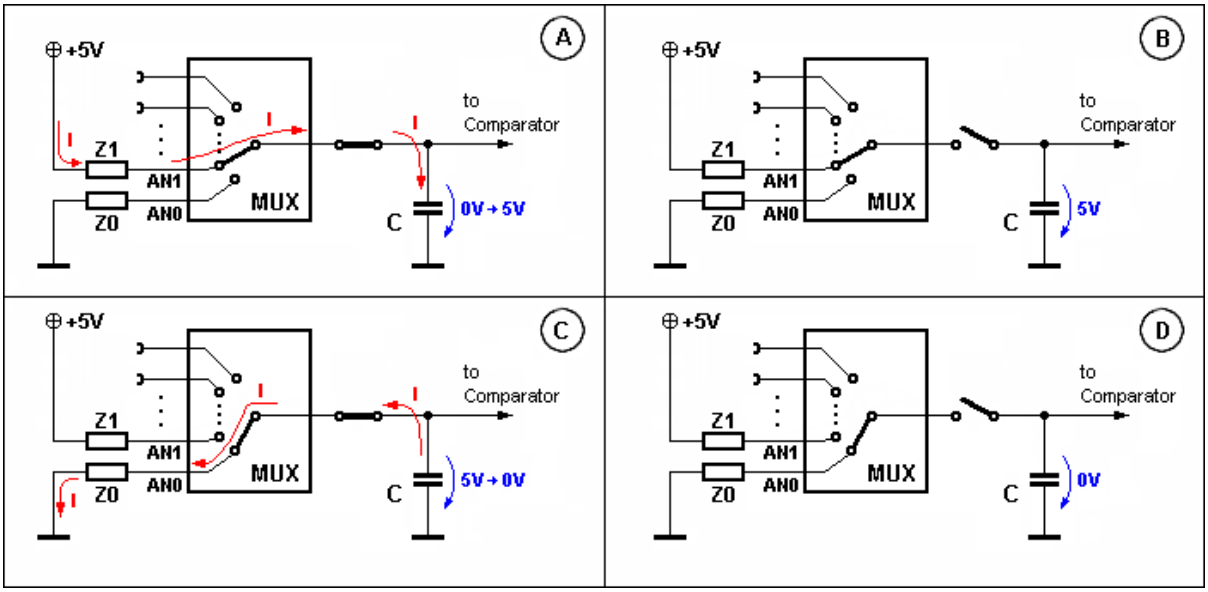
7.1 Recharging and discharging the Sampling capacitor

Because the ADC uses a sampling capacitor the input impedance must be set to a value below 15 kΩ to recharge or discharge this capacitor within the sampling time.

Example:

Assume an application uses two ADC inputs. At one pin there is a voltage of about Vcc and at the other pin Vss. The first conversion charges the internal capacitor to Vcc within the sample time (A). After this the conversion starts (B). At the second conversion the MUX switches to the other input and the capacitor is discharged to Vss (C). The second conversion starts (D). If the input impedance (Z_0) is too high, a rest of charge will remain and a wrong conversion will result.

Figure 13. Current flow in the example



The left illustration shows the voltage glitch on AN0 if Z_0 is low.

The glitch occurs, when the sampling switch is closed.

Figure 14. Voltage glitch at low input impedance

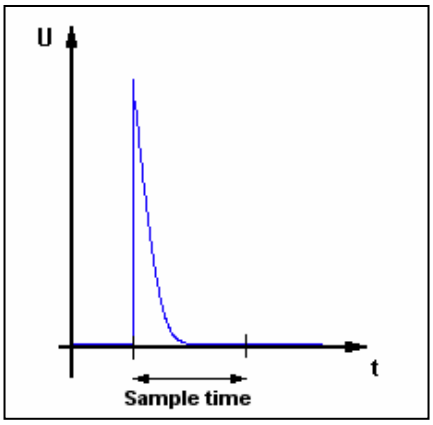
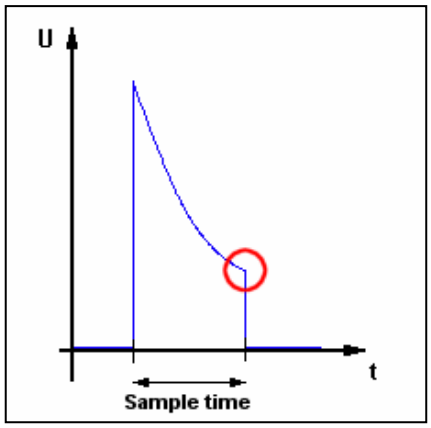


Figure 15. Voltage glitch at high input impedance



In this illustration the input impedance Z_0 is too high. The sampling capacitor is not discharged within the sampling time and thus a wrong voltage is converted (red circle).

Note: At high impedance input circuit, there will always be a glitch of about 0.5 Volts max even if the previous conversion was at 0 Volt. This results from the ADC internal architecture itself.

For the same reason discussed in this chapter an EMI capacitors for any analogue input pins should not exceed 1 nF.

8 ADC Example

Examples for the ADC

8.1 ADC with interrupts

Main.c

```
void InitADC (void)
{
       ADER0 = 0x02;        /* channel 1 to analog input             */
       ADCSL = 0xA0;        /* ADC set to 8 bit continuous mode      */
       ADSR = 0xFC21;       /* ADSRL + ADSRH, Sampling time - 128 CLKP1 Cycles,
                               Conversion time - 264 CLKP1 Cycles, convert only
                               AN1 */
       ADCSH = 0xA2;        /* interrupt enabled, software trigger,
                               start conversion */
}

void main (void)
{
       DDR00 = 0xFF ;       /* Data direction Port 0 = Output        */
       InitIrqLevels();
       __set_il(7);         /* allow all levels                      */
       __EI();              /* globally enable interrupts            */

       InitADC();           /* init AD - converter                   */

       while(1)             /* waiting for interrupt (no operation)  */
             __asm("\tnop");
}

__interrupt void ISR_ADC (void)
{
       PDR00 = ADCRL;       /* shows voltage on Port 0 ( LEDs )      */
       ADCSH = 0xA2;        /* clear interrupt flag                  */
}
```

The above sample code demonstrates to configure ADC in the continuous mode with 8 bit conversion resolution with interrupts enabled. The maximum possible sampling and conversion time is used here. After every ADC interrupt the converted data is output to the LEDs connected to Port 0.

vectors.c

```
void InitIrqLevels (void)
{
        . . .
  ICR = (76 << 8) | 6;     /* Priority Level 6 for ADC of MB96340 Series */

        . . .
}

/* ISR prototype */
__interrupt void ISR_ADC (void);

        . . .

#pragma intvect ISR_ADC 76        /* ADC of MB96340 Series */
        . . .
```

9 Additional Information

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

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

The software examples related to this application note is:

96340_adc

96340_adc_rlt

96340_adc_dma

96340_ppg16_rlt_adc_dma

It can be found on the following Internet page:

http://www.cypress.com/16lx

10 Document History

Document Title: AN205553 - F²MC-16FX Family, A/D Converter

Document Number: 002-05553

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	-	NOFL	07/28/2006	Initial release
			03/01/2007	Reviewed the document and updated with review findings
			09/04/2007	Updated the entire documents with review findings from PHu
*A	5080483	NOFL	03/07/2016	Migrated Spansion Application Note MCU-AN-300215-E-V12 to Cypress format
*B	5846435	AESATP12	08/24/2017	Updated logo and copyright.

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

ARM® Cortex® Microcontrollers	cypress.com/arm
Automotive	cypress.com/automotive
Clocks & Buffers	cypress.com/clocks
Interface	cypress.com/interface
Internet of Things	cypress.com/iot
Memory	cypress.com/memory
Microcontrollers	cypress.com/mcu
PSoC	cypress.com/psoc
Power Management ICs	cypress.com/pmic
Touch Sensing	cypress.com/touch
USB Controllers	cypress.com/usb
Wireless Connectivity	cypress.com/wireless

PSoC® Solutions

PSoC 1 | PSoC 3 | PSoC 4 | PSoC 5LP | PSoC 6

Cypress Developer Community

Forums | WICED IOT Forums | Projects | Videos | Blogs | Training | Components

Technical Support

cypress.com/support

All other trademarks or registered trademarks referenced herein are the property of their respective owners.

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709

© Cypress Semiconductor Corporation, 2006-2017. This document is the property of sCypress Semiconductor Corporation and its subsidiaries, including Spansion LLC ("Cypress"). This document, including any software or firmware included or referenced in this document ("Software"), is owned by Cypress under the intellectual property laws and treaties of the United States and other countries worldwide. Cypress reserves all rights under such laws and treaties and does not, except as specifically stated in this paragraph, grant any license under its patents, copyrights, trademarks, or other intellectual property rights. If the Software is not accompanied by a license agreement and you do not otherwise have a written agreement with Cypress governing the use of the Software, then Cypress hereby grants you a personal, non-exclusive, nontransferable license (without the right to sublicense) (1) under its copyright rights in the Software (a) for Software provided in source code form, to modify and reproduce the Software solely for use with Cypress hardware products, only internally within your organization, and (b) to distribute the Software in binary code form externally to end users (either directly or indirectly through resellers and distributors), solely for use on Cypress hardware product units, and (2) under those claims of Cypress's patents that are infringed by the Software (as provided by Cypress, unmodified) to make, use, distribute, and import the Software solely for use with Cypress hardware products. Any other use, reproduction, modification, translation, or compilation of the Software is prohibited.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS DOCUMENT OR ANY SOFTWARE OR ACCOMPANYING HARDWARE, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. To the extent permitted by applicable law, Cypress reserves the right to make changes to this document without further notice. Cypress does not assume any liability arising out of the application or use of any product or circuit described in this document. Any information provided in this document, including any sample design information or programming code, is provided only for reference purposes. It is the responsibility of the user of this document to properly design, program, and test the functionality and safety of any application made of this information and any resulting product. Cypress products are not designed, intended, or authorized for use as critical components in systems designed or intended for the operation of weapons, weapons systems, nuclear installations, life-support devices or systems, other medical devices or systems (including resuscitation equipment and surgical implants), pollution control or hazardous substances management, or other uses where the failure of the device or system could cause personal injury, death, or property damage ("Unintended Uses"). A critical component is any component of a device or system whose failure to perform can be reasonably expected to cause the failure of the device or system, or to affect its safety or effectiveness. Cypress is not liable, in whole or in part, and you shall and hereby do release Cypress from any claim, damage, or other liability arising from or related to all Unintended Uses of Cypress products. You shall indemnify and hold Cypress harmless from and against all claims, costs, damages, and other liabilities, including claims for personal injury or death, arising from or related to any Unintended Uses of Cypress products.

Cypress, the Cypress logo, Spansion, the Spansion logo, and combinations thereof, WICED, PSoC, CapSense, EZ-USB, F-RAM, and Traveo are trademarks or registered trademarks of Cypress in the United States and other countries. For a more complete list of Cypress trademarks, visit cypress.com. Other names and brands may be claimed as property of their respective owners.