FMEMCU-AN-900084-11

F²MC-16LX FAMILY

16-BIT MICROCONTROLLER

ALL SERIES

A/D CONVERTER

APPLICATION NOTE

Revision History

Date	Issue
18.March 2003	V1.0 (QZ), First release
02. June 2003	V1.1 (QZ), Power consumption measurement
02. June 2003	V1.1 (HW), Reflection of different modes (software, timing diagrams)

This document contains 29 pages.

Warranty and Disclaimer

To the maximum extent permitted by applicable law, Fujitsu Microelectronics Europe GmbH restricts its warranties and its liability for **all products delivered free of charge** (eg. software include or header files, application examples, target boards, evaluation boards, engineering samples of IC's etc.), its performance and any consequential damages, on the use of the Product in accordance with (i) the terms of the License Agreement and the Sale and Purchase Agreement under which agreements the Product has been delivered, (ii) the technical descriptions and (iii) all accompanying written materials. In addition, to the maximum extent permitted by applicable law, Fujitsu Microelectronics Europe GmbH disclaims all warranties and liabilities for the performance of the Product and any consequential damages in cases of unauthorised decompiling and/or reverse engineering and/or disassembling. **Note, all these products are intended and must only be used in an evaluation laboratory environment**.

1. Fujitsu Microelectronics Europe GmbH warrants that the Product will perform substantially in accordance with the accompanying written materials for a period of 90 days form the date of receipt by the customer. Concerning the hardware components of the Product, Fujitsu Microelectronics Europe GmbH warrants that the Product will be free from defects in material and workmanship under use and service as specified in the accompanying written materials for a duration of 1 year from the date of receipt by the customer.

2. Should a Product turn out to be defect, Fujitsu Microelectronics Europe GmbH´s entire liability and the customer´s exclusive remedy shall be, at Fujitsu Microelectronics Europe GmbH´s sole discretion, either return of the purchase price and the license fee, or replacement of the Product or parts thereof, if the Product is returned to Fujitsu Microelectronics Europe GmbH in original packing and without further defects resulting from the customer´s use or the transport. However, this warranty is excluded if the defect has resulted from an accident not attributable to Fujitsu Microelectronics Europe GmbH, or abuse or misapplication attributable to the customer or any other third party not relating to Fujitsu Microelectronics Europe GmbH.

3. To the maximum extent permitted by applicable law Fujitsu Microelectronics Europe GmbH disclaims all other warranties, whether expressed or implied, in particular, but not limited to, warranties of merchantability and fitness for a particular purpose for which the Product is not designated.

4. To the maximum extent permitted by applicable law, Fujitsu Microelectronics Europe GmbH´s and its suppliers´ liability is restricted to intention and gross negligence.

   NO LIABILITY FOR CONSEQUENTIAL DAMAGES

   To the maximum extent permitted by applicable law, in no event shall Fujitsu Microelectronics Europe GmbH and its suppliers be liable for any damages whatsoever (including but without limitation, consequential and/or indirect damages for personal injury, assets of substantial value, loss of profits, interruption of business operation, loss of information, or any other monetary or pecuniary loss) arising from the use of the Product.

Should one of the above stipulations be or become invalid and/or unenforceable, the remaining stipulations shall stay in full effect

Contents

1 Introduction

This application-note gives a rough overview about the embedded A/D-converter (ADC) used within the Fujitsu 16LX series and reflects the different modes that can be used. Design hints are given as well as software examples.

Note:

If not other stated, this documentation is based on MB90390series. However, principles are the same for all Fujitsu 16LX devices.

2 Power supply of A/D converter

2.1 Power consumption

The power consumption (I_R, I_A) of the ADC increases in case a conversion is in progress (ADCS1_BUSY = 1). While the ADC is halted (ADCS1_BUSY = 0), only a leakage current (I_{RH}, I_{AH}) occurs. The following diagrams reflect this behaviour:

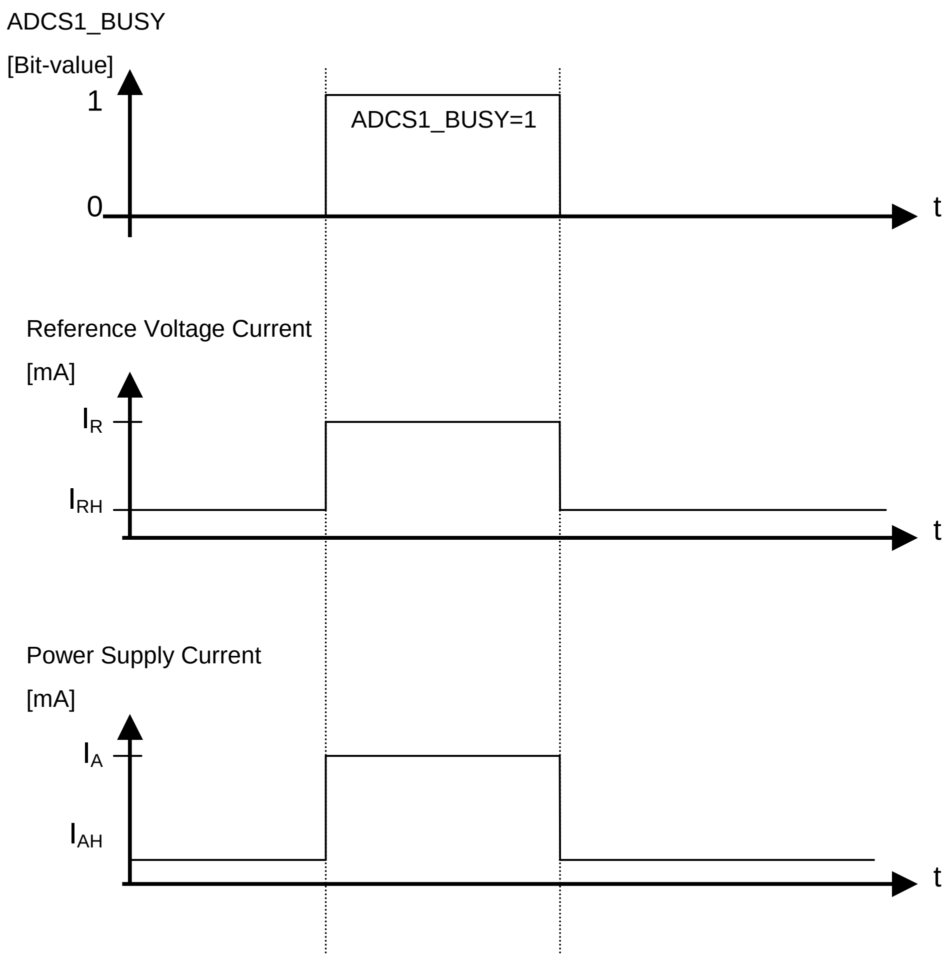


Figure 2-1: Power consumption and operating status of ADC

Note:

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

2.2 Noise consideration

Fujitsu 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 U_{REF}=5V. Hence, the noise introduced from the external circuits must considered and should be reduced to the minimum as possible.

The reference voltage U_{REF}, which is equal to AVRH-AVRL, is connected to the weighted capacitor array and the resistor array of the ADC. The noise coupled to U_{REF} 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 U_{REF} must have a very low AC impedance. For this purpose, an external bypass capacitor should be applied. In praxis two capacitors in parallel are recommended, one filtering low frequency noise, the other one filtering high frequency noise ((100nF–10µF)||(10pF-00pF)). 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. V_{CC} is very noise. For this reason, the pins AV_{CC} and V_{CC} should be not short circuit directly. Instead, the both paths should be routed to the sources separately. Two additional capacitors should be located close to both pins as de-coupling capacitors. For more efficient noise filtering the same configuration as for AVRH is recommended.

If a π mode RC filter is used, the analogue supply current must be taken into account, which might cause additional voltage drop within the π mode RC filter. Generally, on the PCB board all bypass capacitors should be put near the pins as close as possible to avoid the side effect of the wiring inductance and resistance of the capacitors.

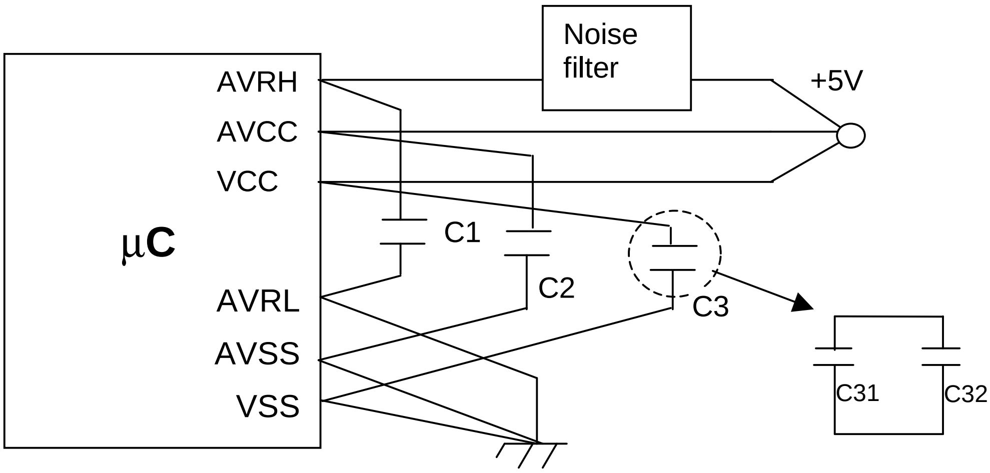


Figure 2-2: A suggested connection for the power supplies

3 Analogue input and related external circuits

3.1 External circuits for analogue input

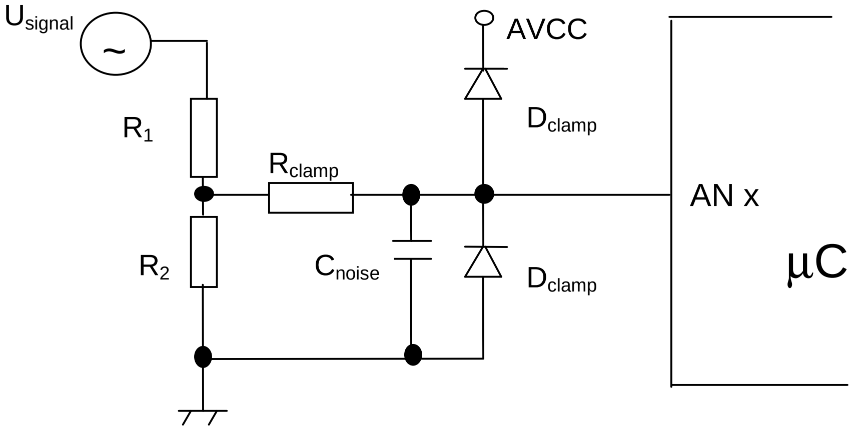


Figure 3-1: A typical external circuit for analogue input

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 3-1).

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 influence 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 $(R_{clamp}+R_1//R_2)xC_{noise}$ should be chosen considerable larger than the sampling time (5 to10 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.

3.2 Input Leakage current consideration

The analogue input pins shows a small leakage current, whose maximum value is ranged from 5µA down to 0.3µA for the new technology. 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 $R_1//R_2+R_{clamp}$ should be not chosen too large according the following equation.

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

Note:

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

To keep the error smaller than one LSB for a leakage of 5µA, the size of $R_1//R_2+R_{clamp}$ should be smaller than 1KΩ. As the leakage current drops down to 0.3 µA. the value of $R1//R2+R_{clamp}$ can be chosen as large as 16KΩ.

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 multiplex 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.

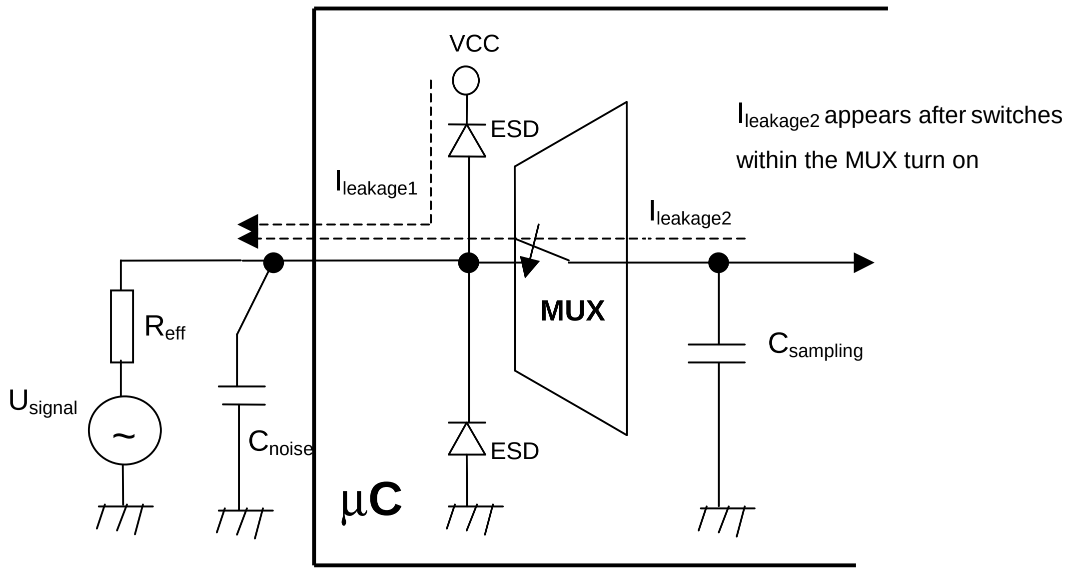


Figure 3-2: Leakage current flowing to the analogue input pin

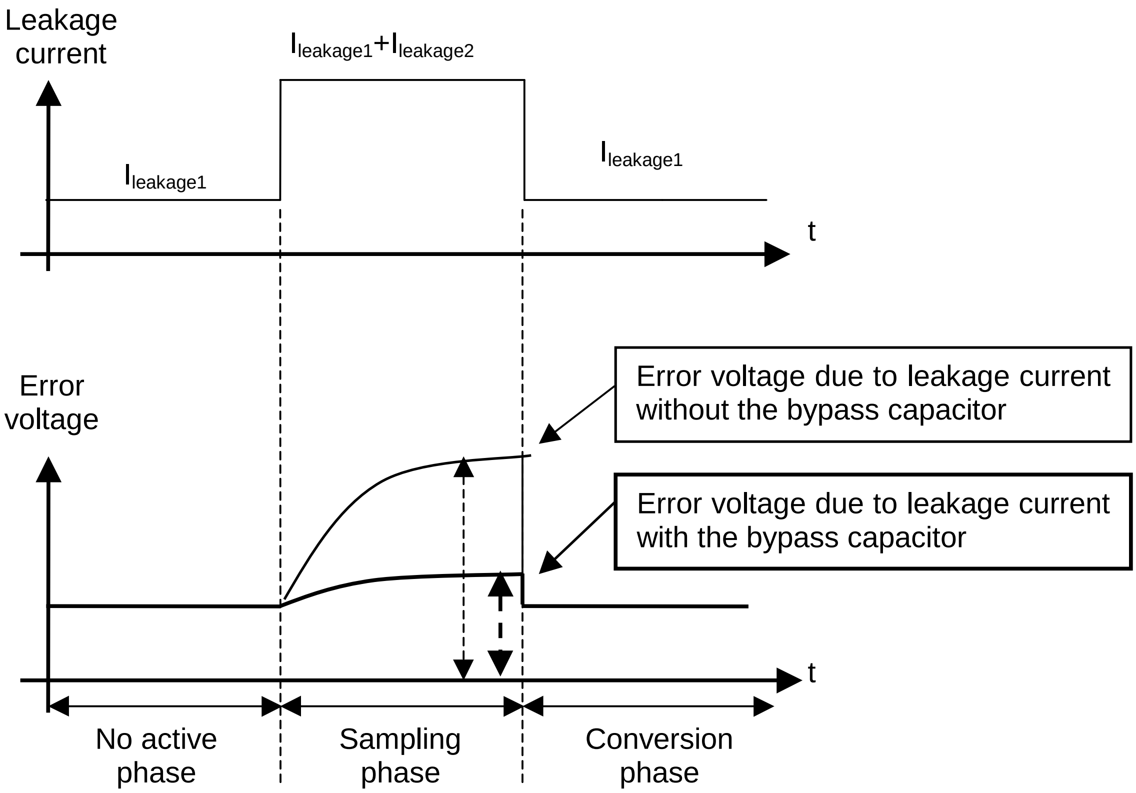


Figure 3-3: Reducing the leakage current with the bypass capacitor

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 5µA 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:

$$C = \frac{5\mu A \times 5\mu s}{4.9mV/2} \approx 10nF$$

4 Sampling time consideration

Fujitsu 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 charger 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 9 of the RC constant, namely, $9 \times (R_{extern} + R_{switch}) \times C_{sampling}$. Then the error amounts to $e^{-9} \times U_{REF}$, corresponding $0.13 \times U_{LSB}$ only.

For MB90340 series, the on-resistor of the transmission gate amounts to 3.4KΩ and the sampling capacitor $C_{sampling}$ 12.4pF. For an external resistor of 3.4kΩ, the sampling time should be chosen larger than $9 \times (3.4k\Omega + 3.4k\Omega) \times 12.4pF = 0.76\mu s$ by using above thumb rule.

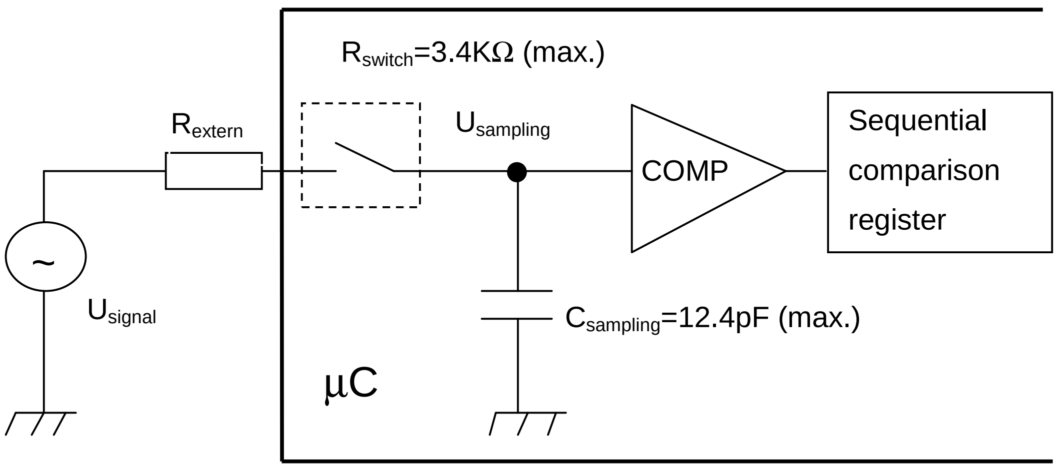


Figure 4-1: Block diagram for ADC in MB90F340

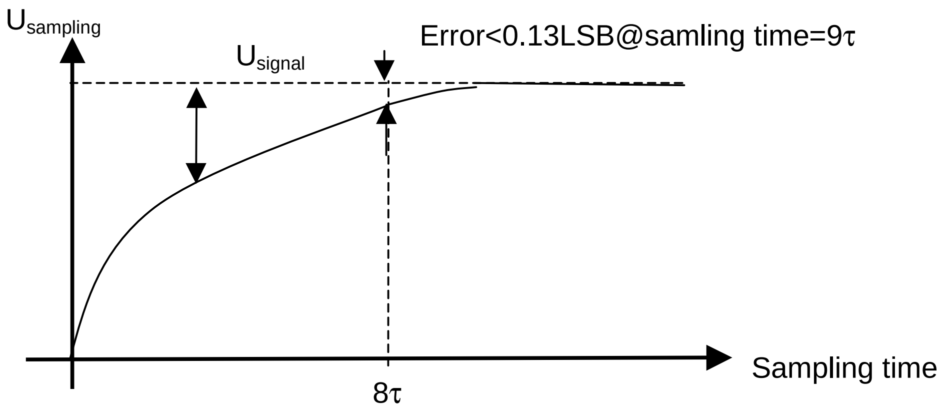


Figure 4-2: Error related to sampling time

For the Fujitsu microcontroller the sampling time can be set by defining the register bit ST2 to ST0 of the ADC data register (ADCR1). Please refer to the hardware manual of the corresponding devices.

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

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.

5.1 $AV_{CC} > V_{CC}$

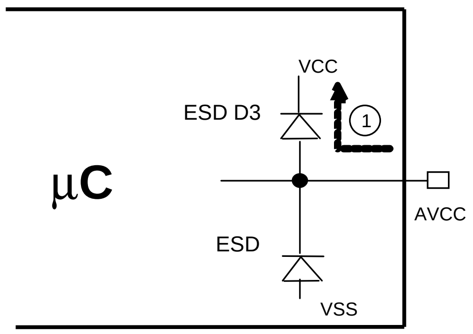


Figure 5-1: Latch-up in case $AV_{CC} > V_{CC}$

Latch-up can happens, 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.

5.2 $AV_{CC} < V_{CC}$

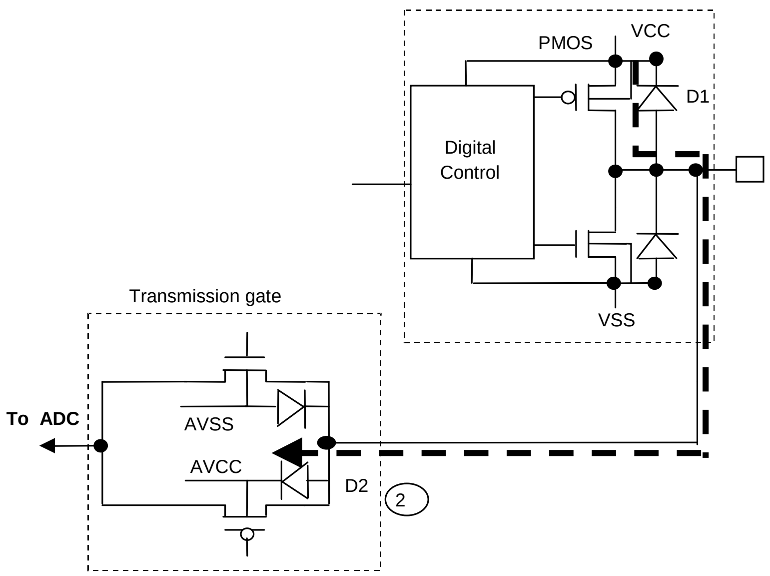


Figure 5-2: Latch-up in case $AV_{CC} < V_{CC}$

This case happens if the analogue pin is used as a digital output pin, the output level is "H", and at the same time, AV_{CC} 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 be not expected at the output, due to the load diode D2.

5.3 U_{AIN} > AV_{CC} or V_{CC}

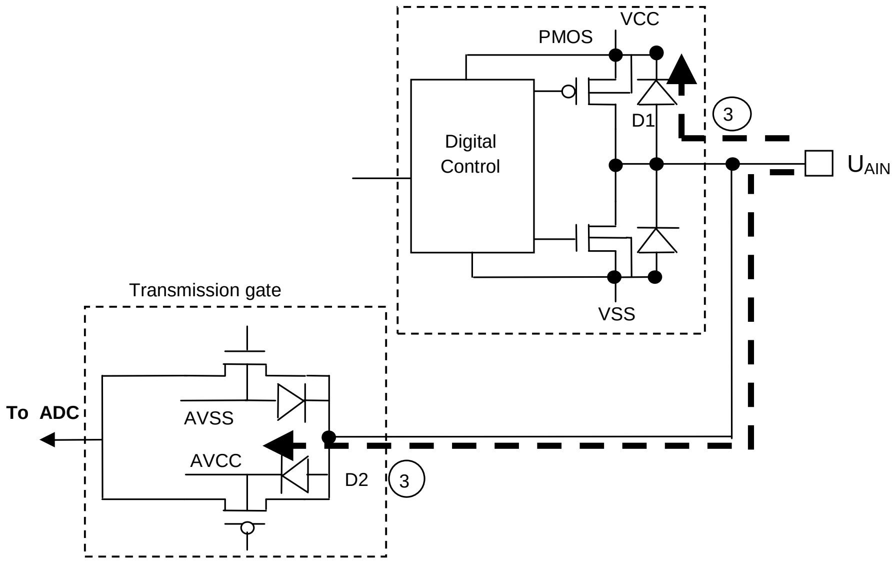


Figure 5-3: Problem in case V_{AIN} > AVCC or V_{CC}

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}, can not be converted by the ADC properly.

5.4 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.

6 Software

This chapter explains the register model of the A/D-converter and gives software-examples how to use the different modes

6.1 Overview

Although different ADC-macros exist within Fujitsu 16LX-family the basics are the same, and the following examples can be adapted easily from one microcontroller series to another. Please refer to the hardware-manual of the corresponding device to get more details.

Figure 6-1 shows a simplified block diagram of the A/D-converter.

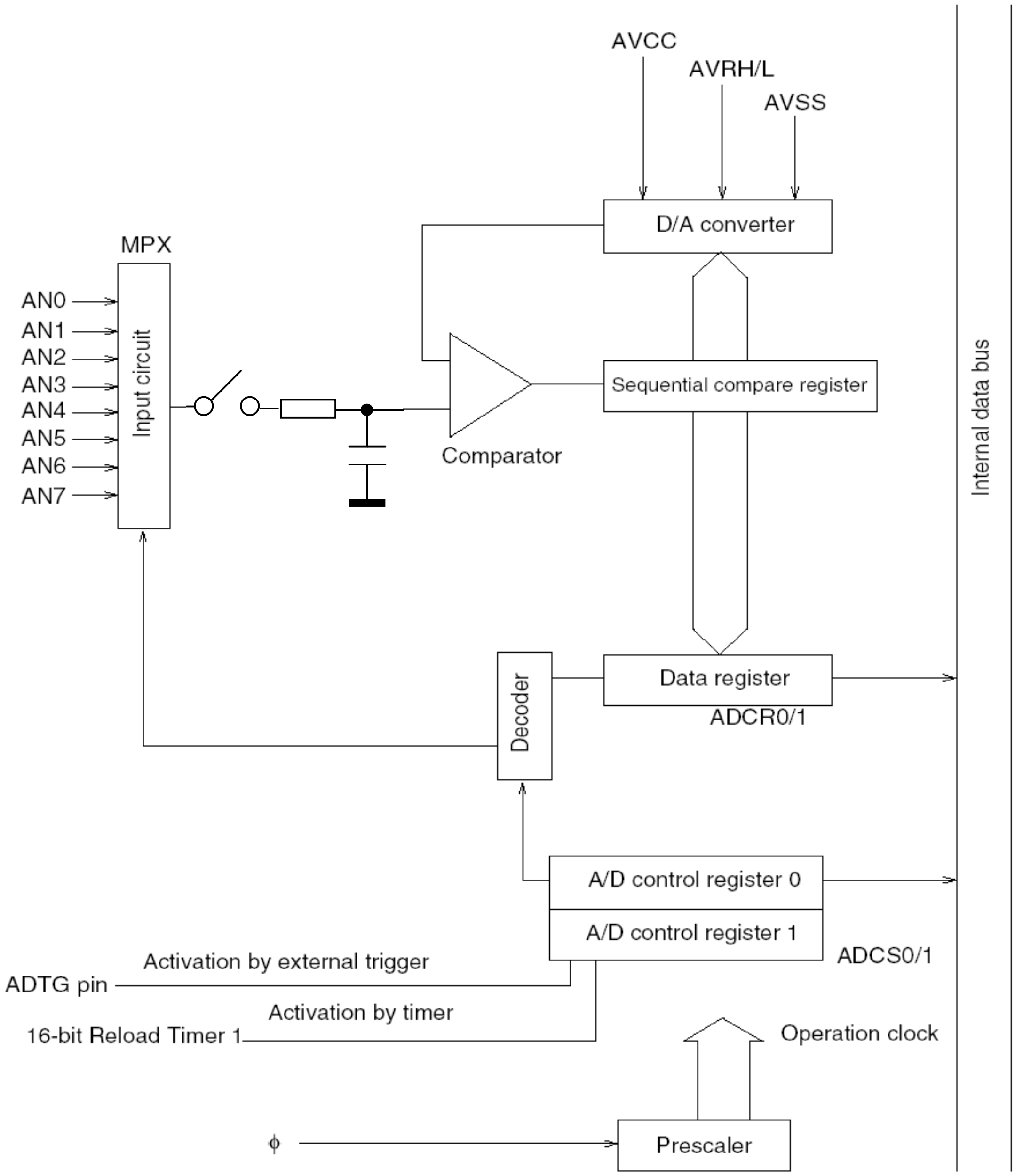


Figure 6-1: Block diagram of an A/D-converter

Four registers exist to define the function and to store the result of the ADC.

ADCS0, ADCR1, ADCS0 and ADCS1

Note: Do not use Bit-operations for all ADC-registers!

6.1.1 ADCS0

The register ADCS0 selects the conversion mode and conversion channel.

bit7 Register: ADCS0 bit0

MD1	MD0	ANS2	ANS1	ANS0	ANE2	ANE1	ANE0
Mode		Start-channel			End-channel		

Mode settings:

0	0	Single conversion mode (retriggerable)
0	1	Single conversion mode (not retriggerable)
1	0	Continuous conversion mode
1	1	Stop conversion mode

The Start-channel ANS may have different meaning while read- and write-access:

Writing to ANS defines the first channel that will be converted after the ADC is started.

Reading from ANS will reflect the current channel that is just in progress. In the Stop conversion mode ANS indicates the channel of the last conversion.

Note: Do not use Bit-operations for writing to this register! Use byte-transfer instead!

6.1.2 ADCS1

The register ADCS1 is the control and status register for the A/D-converter

bit7 Register: ADCS1 bit0

BUSY	INT	INTE	PAUS	STS1	STS0	STRT	'0'

```
|     |     |     |     |     |     '1' starts conversion
|     |     |     |     >     >     '0' don't care
|     |     |     |     0     0     only software trigger
|     |     |     |     0     1     External trigger or software
|     |     |     |     1     0     ReloadTimer1 or software
|     |     |     >     1     1     Ext.Trig or RLT1 or software
|     |     >     PAUS is only valid when EIIOS is used
|     >     '1' enables Interrupt request, '0' disables Interrupt request
>     Reading '1' indicates completion of A/D-conversion, Writing '0' clears this bit
Reading '1' indicates the A/D-conversion is in progress, Writing '0' stops A/D-conversion
```

Note: Do not start (STRT=1) and stop (BUSY=0) at the same time.

Note: Do not use Bit-operations for writing to this register! Use byte-transfer instead!

6.1.3 ADCR1

The register ADCR1 selects the resolution and holds the upper 2-bit of the 10-bit result of the A/D conversion, as well as defines the sampling/comparison time.

Register: ADCR1

bit7							bit0
S10	ST1	ST0	CT1	CT0	-	D9	D8

upper 2 bits of A/D-result

Comparison time (please refer to the corresponding hardwaremanual)

Sampling time (please refer to the corresponding hardwaremanual)

Writing ‘0’ selects 10-bit resolution, Writing ‘1’ selects 8-bit resolution

Note: Do not use Bit-operations for writing to this register! Use byte-transfer instead!

6.1.4 ADCR0

The register ADCR0 holds the lower 8-bit of the result of the A/D conversion

Register: ADCR0

bit7							bit0
D7	D6	D5	D4	D3	D2	D1	D0

8-bit A/D-result (lower 8-bit if 10-bit resolution is selected)

A byte-transfer should be used to read the 8-bit result of an A/D-conversion:

```
unsigned char result;
result = ADCR0; // read 8-bit result
```

A word-transfer should be used to read the 10-bit result of an A/D-conversion:

```
unsigned int result;
result = ADCR; // read 10-bit result
```

6.1.5 ADER

Additional a register ADERx may be available to switch the functionality of a port between I/O-port and Analogue-port.

Register: ADERx

bit7							bit0
ADE7	ADE6	ADE5	ADE4	ADE3	ADE2	ADE1	ADE0

Setting the bit to ‘1’ enables the corresponding port-pin as Analogue-input.

Setting the bit to ‘0’ enables the corresponding port-pin as digital I/O-port.

Generally, these bits are initialised with ‘1’.

Please refer to the hardware-manual of the corresponding device to get more details.

6.2 Stop Mode

Register: ADCS0

bit7			bit0
1	1	ANS: Start-channel	ANE: End-channel

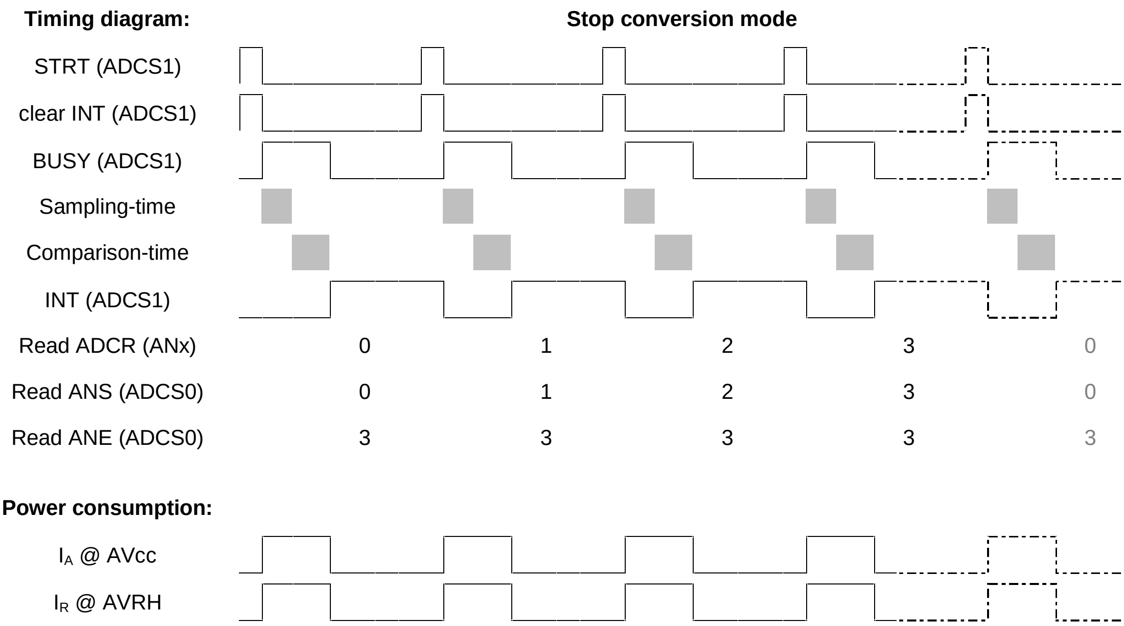
The stop conversion mode is the simplest mode. Beginning from the start-channel (ANS) each channel is converted successively until the stop-channel (ANE) is reached. Then the conversion begins again from the start-channel. After each conversion is completed ADCS1_BUSY is cleared and ADCS1_INT is set. To continue conversion with the next channel the start-condition defined by ADCS1_ST1/ST0 has to be met. In case that ANS=ANE only one channel is converted repetitively.

6.2.1 Stop Mode: Example

```
unsigned int result; // stores the result of A/D-conversion
void main(void)
{
  ADER0 = 0xFF; // Analog input enable
  ADCR1 = 0xE8; // Resolution, Sampling Time, Conversion Time
  ADCS0 = 0xC3; // Mode: Stop conversion mode, start-channel 0, end-channel 3
  ADCS1 = 0x82; // Control: disable interrupt, clear INT-Flag, mode software activation, START
  while(1) {
    if (ADCS1_INT) {
      result = ADCR & 0x03ff; // read 10-bit result
      //result = ADCR0;       // read 8-bit result
      ADCS1 = 0x82;           // clear INT, START-trigger for next channel
    }
  }
}
```

6.2.2 Stop Mode: Timing diagram

e.g. ANS=0, ANE=3



6.2.3 Stop Mode: Power consumption measurements

Please refer to chapter 7.1 (Supply current measuring) for details of measuring.

The measurement in Figure 6-2 shows the voltage drop U_{AVCC} of the shunt resistor R_A and the voltage drop U_{AVRH} at the shunt resistor R_R while a conversion is done. Easily can be seen that the supply current flows only while converting.

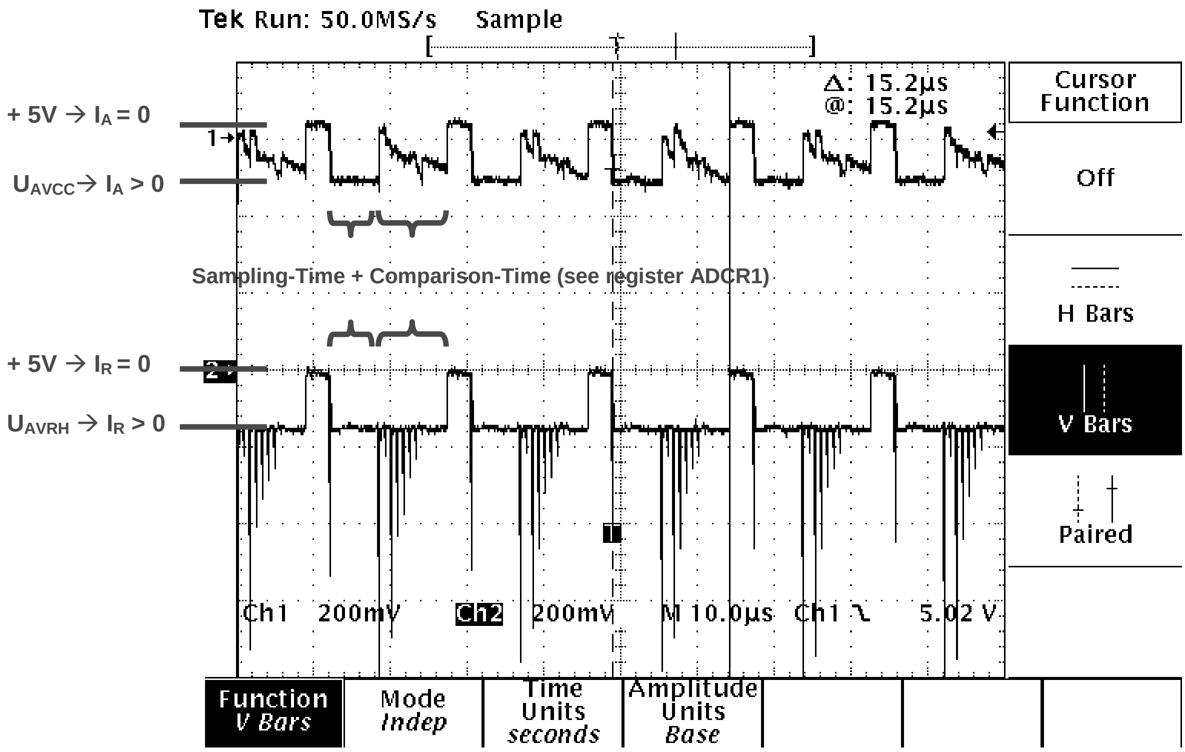


Figure 6-2: Power consumption in stop conversion mode

6.3 Single Conversion Mode 1

bit7 Register: ADCS0 bit0

0	0	ANS: Start-channel	ANE: End-channel

Once started the A/D-converter in the single conversion mode 1 automatically one conversion round from start-channel to end-channel without any pause is done. Conversion stops after the last channel (ANE). In single conversion mode 1, a restart by setting ADCS1_STRT while processing is allowed.

The single conversion mode 1 should always be used together with an interrupt service routine for the ADC-interrupt. As soon as the interrupt enable bit (ADCS1_INTE) is set the internal data protection function will be prevent the data register from overwriting. Clearing the ADCS1_INT bit enables next conversion.

6.3.1 Single Conversion Mode 1: Example

```
unsigned int result; // stores the result of A/D-conversion

__interrupt void IRQ_ADC (void)
{
  result = ADCR0;           // read 8-bit result
  //result = ADCR & 0x03ff; // read 10-bit result
  ADCS1 = 0xA0;             // clear INT-Flag
}

void main(void)
{
  InitIrqLevels();
  __set_il(7);      // allow all levels
  __EI();           // globaly enable interrupts

  ADER0 = 0xFF;     // Analog input enable
  ADCR1 = 0xE8;     // Resolution, Sampling Time, Conversion Time
  ADCS0 = 0x03;     // Mode: Single conversion mode 1, start-channel 0, end-channel 3
  ADCS1 = 0xA2;     // Control: INT-enable, clear INT-Flag, Start conversion
  ICR10 = 2;        // set IRQ-Level-Register: IRQ31 (ADC), IRQ32 (IO-Timer)

  while(1)
  {
    asm("\tNOP");   // nothing to do, everything is done by interrupts
  }
}
```

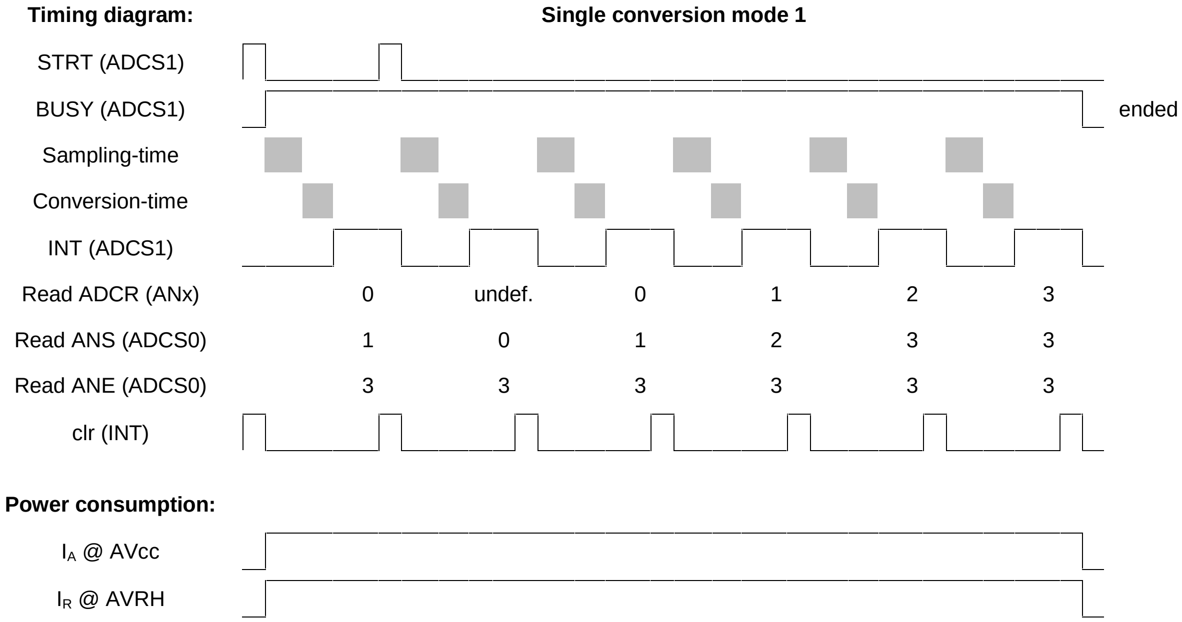
Note: Do no forget to define the Interrupt-Service-Routine in `vector.c` !
e.g.: MB90390:

```
    __interrupt void DefaultIRQHandler (void);
--> __interrupt void IRQ_ADC (void);

    #pragma intvect DefaultIRQHandler 30    /* Output Capture 4/5                 */
--> #pragma intvect IRQ_ADC           31    /* A/D Converter                      */
    #pragma intvect DefaultIRQHandler 32    /* I/O Timer0, I/O Timer1 / Watch Timer */
```


6.3.2 Single Conversion Mode 1: Timing diagram

e.g. ANS=0, ANE=3



6.3.3 Single Conversion Mode 1: Power consumption

The measurement in Figure 6-3 shows the voltage drop U_{AVCC} of the shunt resistor R_A and the voltage drop U_{AVRH} at the shunt resistor R_R while a round of conversion is done. Easily can be seen that the supply current flows from the beginning of the first channel (ANS) until the last channel (ANE) is converted.

Please refer to chapter 7.1 (Supply current measuring) for details.

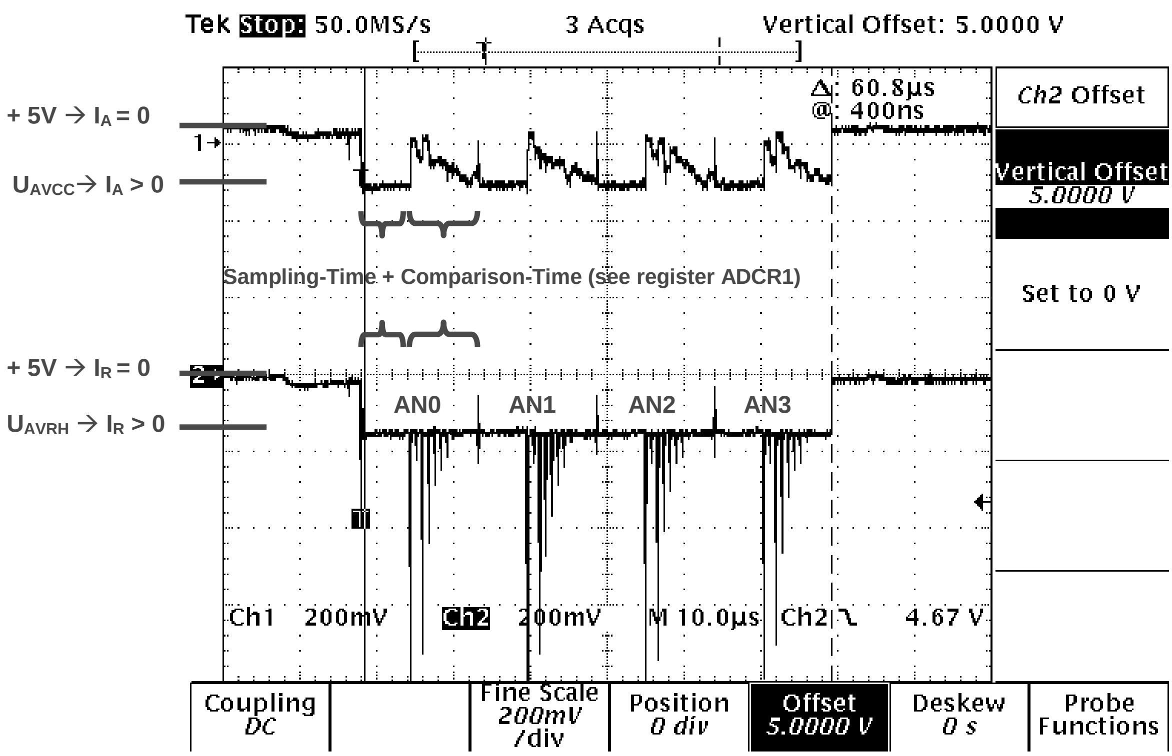


Figure 6-3: Power consumption in single conversion mode

6.3.4 Retriggering of Single Conversion Mode 1

Please take care, that the possibility to restart the ADC should only be used with the software trigger (see register ADCS1: STS1=0, STS0=0).

In case that the restart occurs while the conversion of the next channel is already started, the ongoing conversion will be interrupted. In this case, an undefined value will be placed in the data register (ADCR), but start-channel (ANS) will be reset to show next valid channel number. See also chapter 6.3.2 (Single Conversion Mode 1: Timing diagram).

Therefore, Single Conversion Mode 1 should not be used together with external trigger (ADTG) or reload-timer, in order to prevent from unexpected undefined data.

6.4 Single Conversion Mode 2

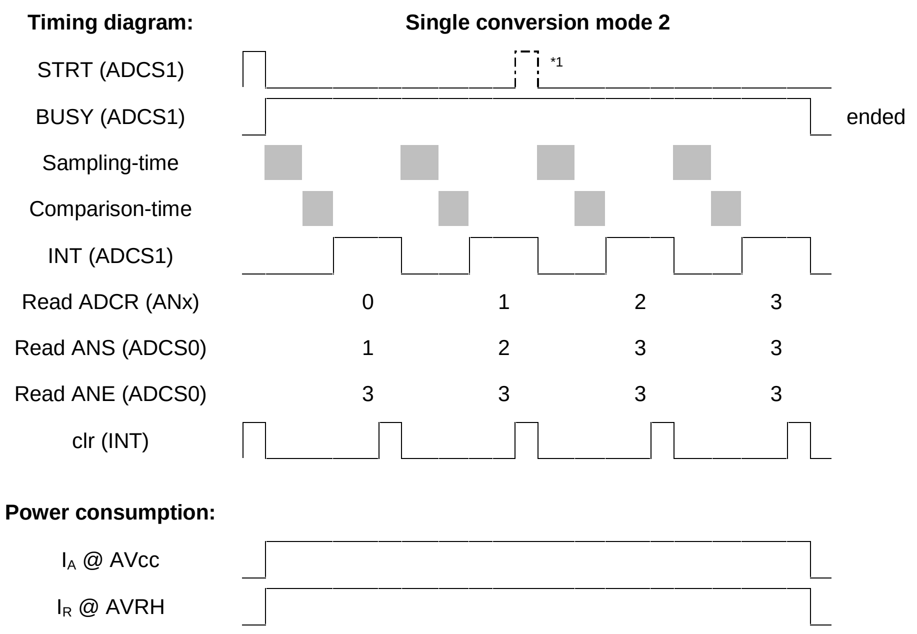
Register: ADCS0

bit7			bit0
0	1	ANS: Start-channel	ANE: End-channel

Single conversion mode 2 behaves same as single conversion mode 1 but one exception: In mode 2 the A/D-converter can not be restarted.

6.4.1 Single Conversion Mode 2: Timing diagram

e.g. ANS=0, ANE=3



*1 Restart will not be accepted in single conversion mode 2

6.5 Continuous Conversion Mode

bit7 Register: ADCS0 bit0

1	0	ANS: Start-channel	ANE: End-channel

Once started the A/D-converter in the continuous conversion mode 1, repetitively conversions round from start-channel to end-channel without any pause are done. Conversion never stops automatically.

The continuos conversion mode should always be used together with an interrupt service routine for the ADC-interrupt. As soon as the interrupt enable bit (ADCS1_INTE) is set the internal data protection function will be prevent the data register from overwriting. Clearing the ADCS1_INT bit enables next conversion.

6.5.1 Continuous Conversion Mode: Example

```
unsigned int result; // stores the result of A/D-conversion

__interrupt void IRQ_ADC (void)
{
  result = ADCR0;           // read 8-bit result
  //result = ADCR & 0x03ff; // read 10-bit result
  ADCS1 = 0xA0;             // clear INT-Flag
}

void main(void)
{
  InitIrqLevels();
  __set_il(7);       // allow all levels
  __EI();            // globaly enable interrupts

  ADER0 = 0xFF;      // Analog input enable
  ADCR1 = 0xE8;      // Resolution, Sampling Time, Conversion Time
  ADCS0 = 0x43;      // Mode: Continuous conversion mode, start-channel 0, end-channel 3
  ADCS1 = 0xA2;      // Control: INT-enable, clear INT-Flag, Start conversion
  ICR10 = 2;         // set IRQ-Level-Register: IRQ31 (ADC), IRQ32 (IO-Timer)

  while(1)
  {
    asm("\tNOP");    // nothing to do, everything is done by interrupts
  }
}
```

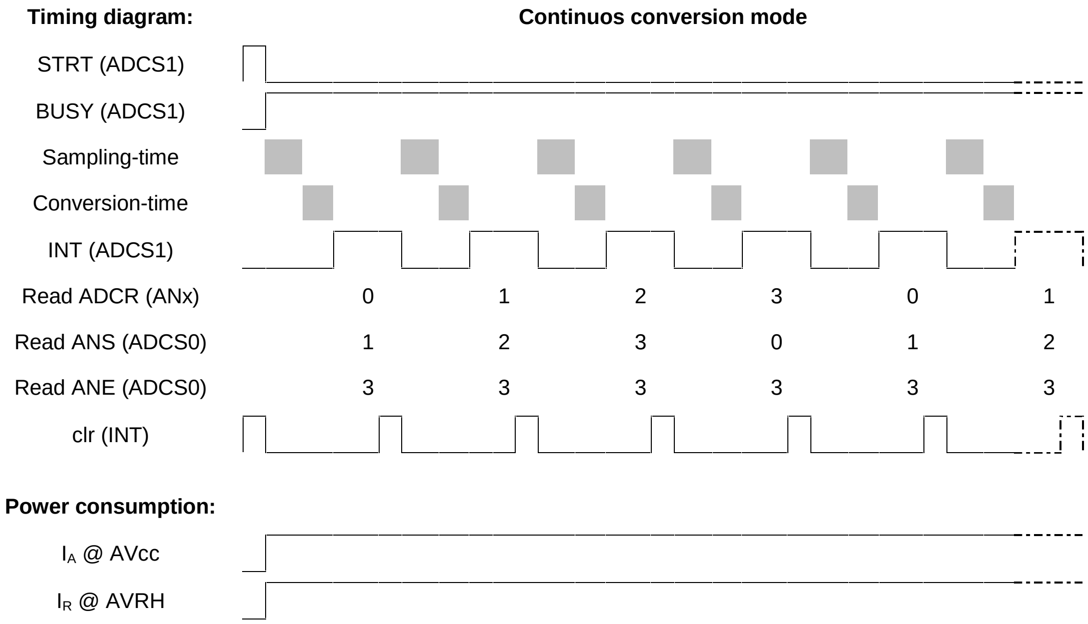
Note: Do no forget to define the Interrupt-Service-Routine in `vector.c` !
e.g.: MB90390:

```
    __interrupt void DefaultIRQHandler (void);
--> __interrupt void IRQ_ADC (void);

    #pragma intvect DefaultIRQHandler 30    /* Output Capture 4/5                 */
--> #pragma intvect IRQ_ADC           31    /* A/D Converter                      */
    #pragma intvect DefaultIRQHandler 32    /* I/O Timer0, I/O Timer1 / Watch Timer */
```

6.5.2 Continuous Conversion Mode: Timing diagram

e.g. ANS=0, ANE=3



6.5.3 Continuos Conversion Mode: Power consumption

The measurement in Figure 6-4 shows the voltage drop U_{AVCC} at the shunt resistor R_A and the voltage drop U_{AVRH} at the shunt resistor R_R. Easily can be seen that the supply current flows all the time after the continuous conversion mode is started. This is, because conversion never ends until stopped manually.

Please refer to chapter 7.1 (Supply current measuring) for details.

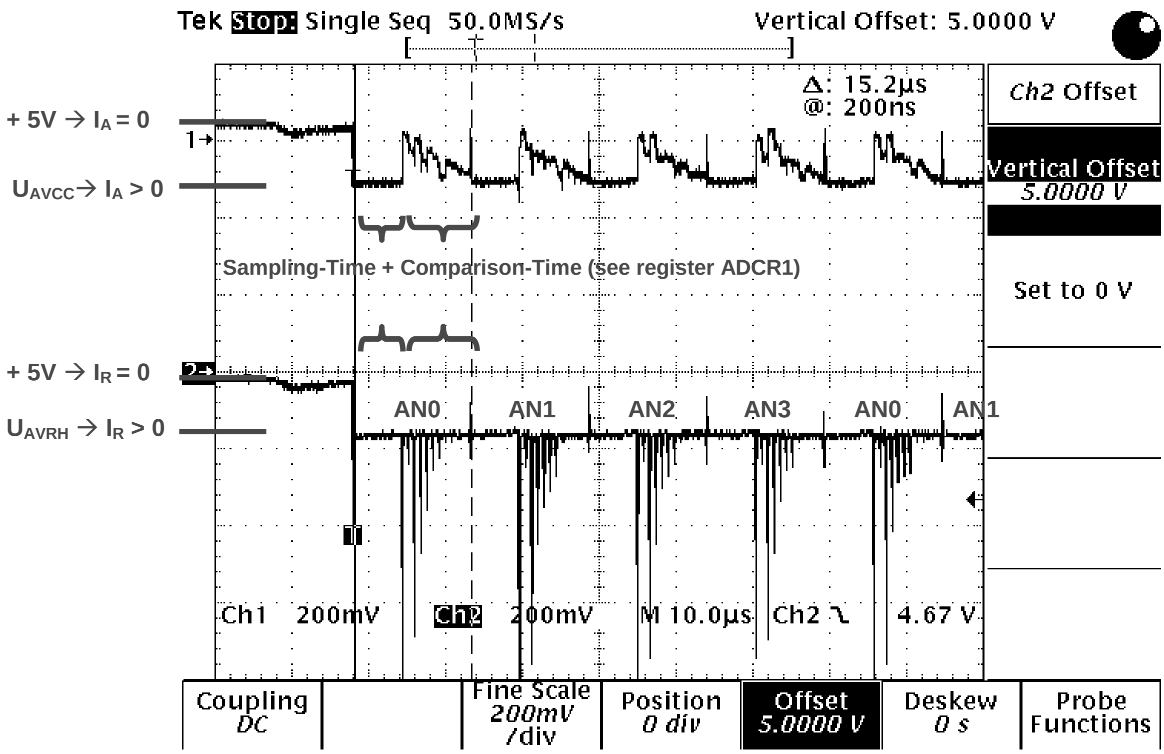


Figure 6-4: Power consumption in continuous conversion mode

6.6 Trigger Modes

Additional to a software-start some devices offer the possibility to start the A/D-conversion by various conditions. The selection has to be done in the control register ADCS1. Please refer to the hardware-manual of the corresponding device to find out what trigger-conditions are offered, e.g. MB90390series:

Register: ADCS1

bit7							bit0
BUSY	INT	INTE	PAUS	STS1	STS0	STRT	'0'

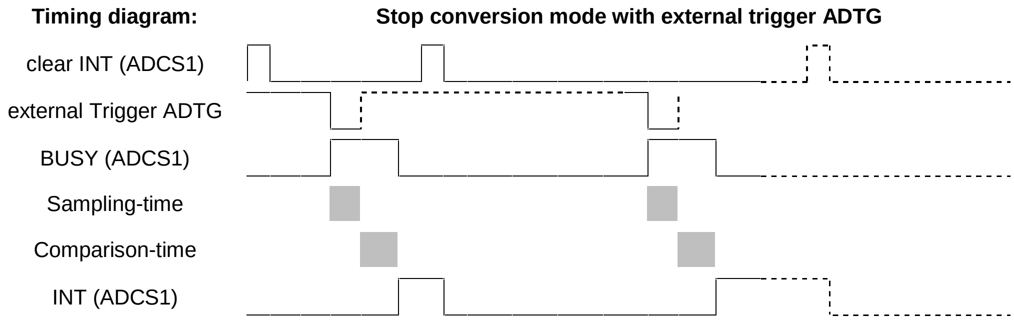
STS1	STS0	
0	0	only software trigger
0	1	External trigger or software
1	0	ReloadTimer1 or software
1	1	Ext.Trig or RLT1 or software

The operation behaviour of the AD-converter depends on the mode setting of register ADCS0 like it is described in chapter 6.2 - 6.5 but with the additional trigger option.

6.6.1 External Trigger Mode (ADTG)

Additional to the software-start the A/D-conversion may be activated by an external trigger signal (ADTG).

```
ADCS1 = 0x84;     // Control: clear INT-Flag, Activation by external trigger ADTG
```



In general, the external trigger reacts on falling edge. Anyhow, please refer to the A/D-converter feature-list within the hardware-manual of the corresponding device to find out the valid edge-level of the external trigger.

6.6.2 Reload Timer

Most devices allow triggering the A/D-converter periodically by the Reload Timer. Please take care that only one dedicated Reload Timer may be connected internally to the A/D-converter. In general, the A/D-converter reacts on the rising edge of the Reload Timer. Anyhow, please refer to the A/D-converter feature-list within the hardware-manual of the corresponding device to find out the valid edge-level of the Reload Timer.

Note: Additionally to the description within the most hardware-manuals, bit OUTL of register TMCSRx may not only influence the output level of the TOT-pin, if the Reload Timer output is enabled (OUTE (TMCRx)= 1), but also the trigger-point of the A/D-converter.

The Reload Timer supports to different modes that will be reflected in the following ensemble with the A/D-converter:

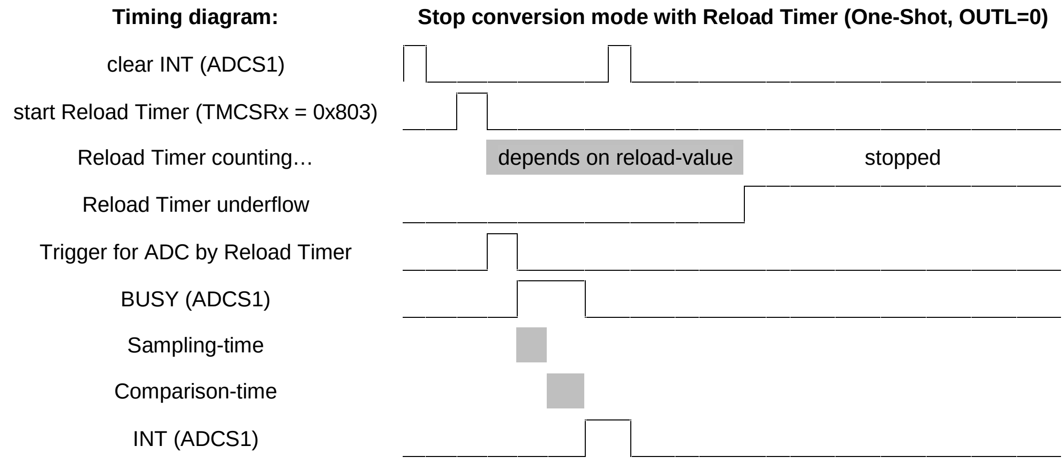
6.6.2.1 One Shot Mode

The One Shot mode may not be such interesting for the A/D-converter because the trigger may occur only one time. This time depends on the setting of bit OUTL of register TMCSRx.

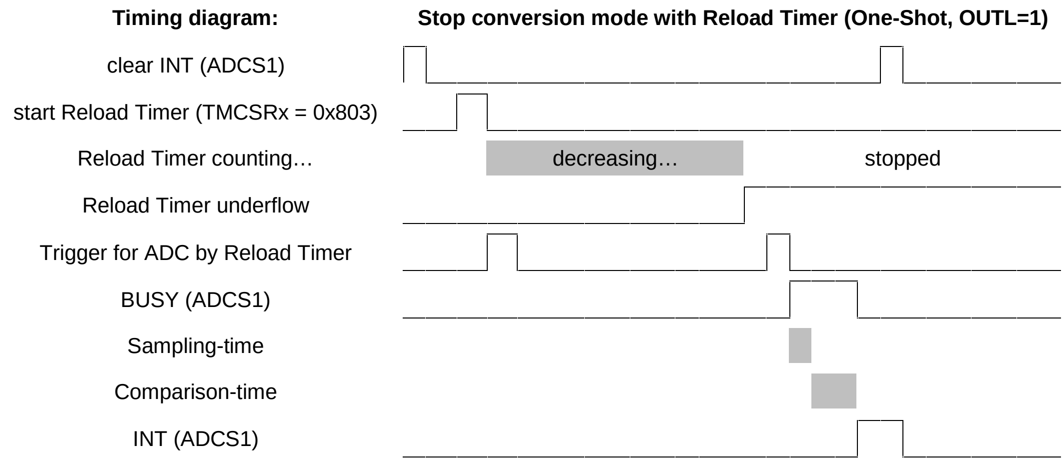
In case that OUTL=0 the reload value of the Reload Timer does not influence the trigger time. The A/D-converter is triggered simultaneously with starting the Reload Timer.

If OUTL=1 then the trigger occurs simultaneously with the underflow of the Reload Timer.

6.6.2.2 OUTL = 0: TMCSRx = 0x803;



6.6.2.3 OUTL = 1: TMCSRx = 0x823;



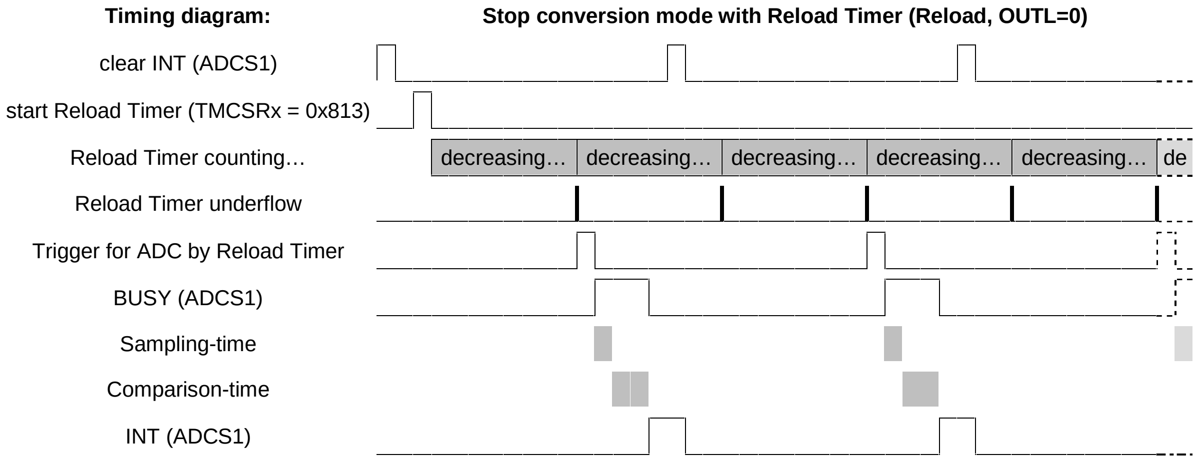
6.6.3 Reload Mode

In Reload mode, the A/D-converter is triggered periodically depending on the reload value. Because the A/D-converter reacts only on the rising edge of the Reload-Timer, the trigger frequency is half the setting of the reload-value, e.g.:

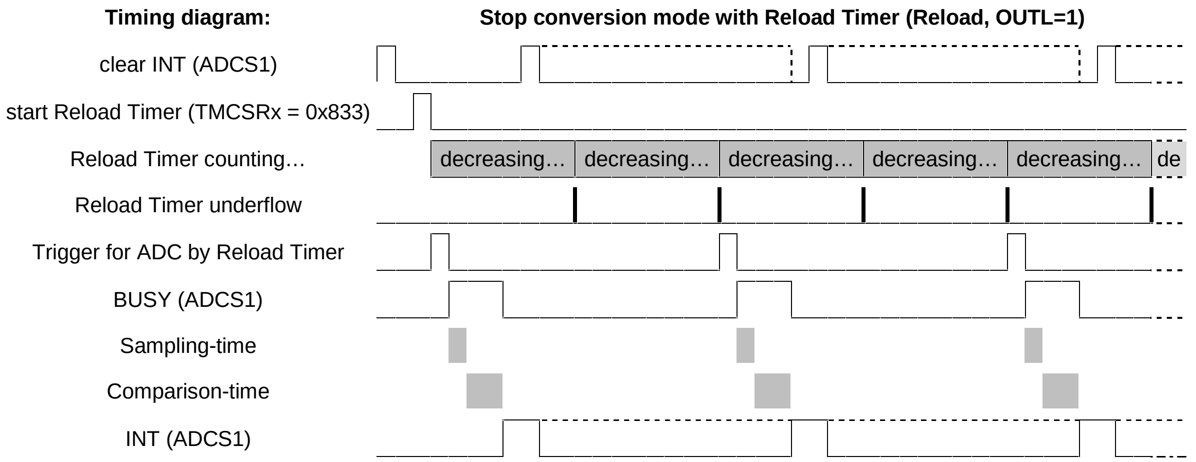
Sampling rate should be 100ms ➔ set Reload Timer to 50ms

Note: The first trigger time depends on the setting of bit OUTL of register TMCSRx.

6.6.3.1 OUTL = 0: TMCSRx = 0x813;



6.6.3.2 OUTL = 1: TMCSRx = 0x833;



6.6.4 16-bit I/O-Timer Mode

Some devices (e.g. MB90560series) offer the possibility to trigger the A/D-converter by the 16-bit Free-Running-Timer (I/O-Timer). Each time the value of the timer is equal to zero the A/D-converter is triggered. This mode is more or less comparable with the Reload-mode and can be used for periodical triggering of the A/D-converter.

7 Appendix

7.1 Supply current measuring

A typical test circuit is shown in Figure 7-1. Two shunt-resistors R_A and R_R are used to calculate the very low currents of I_A@AVcc and I_R@AVRH by the voltage drop:

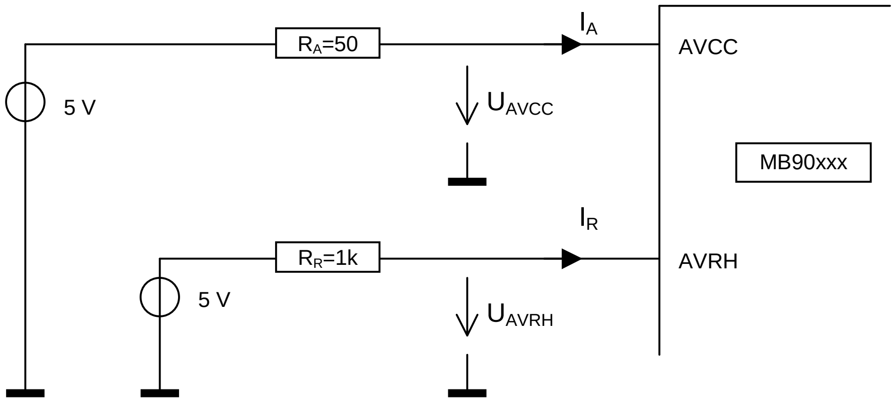


Figure 7-1: Test circuit

$$I_A = \frac{5V - U_{AVCC}}{R_A}$$

$$I_R = \frac{5V - U_{AVRH}}{R_R}$$

By the following measurement (Figure 7-2), that was done with a MB90F394 in stop conversion mode, the calculation of the power consumption will be discussed.

While not converting, no current (I_A, I_R) flows that would cause a voltage drop of R_A and R_R. Therefore, the levels at AVCC and AVRH remain at 5V. While conversion runs the currents (I_A, I_R) cause a voltage drop of R_A and R_R.

From the diagram ΔAVCC=165mV and ΔAVRH=160mV can be read (200mV/div):

$$I_A = \frac{5V - U_{AVCC}}{R_A} = \frac{165mV}{50\,\Omega} = 3.3mA \qquad I_R = \frac{5V - U_{AVRH}}{R_R} = \frac{160mV}{1000\,\Omega} = 160\mu A$$

Within the same measurement (10µs/div) also sampling-time and comparison-time can be checked easily:

$$\text{Sampling Time} = 0.64 \times 10\mu s = 6.4\mu s \qquad \text{Comparison Time} = 0.88 \times 10\mu s = 8.8\mu s$$

The measurement is confirmed by the software settings:

Internal operating clock: 20MHz Internal machine clock cycle: 50ns

Setting of A/D-conversion: ADCR1 = 0xE8;
(8-Bit resolution, SamplingTime:128cycl, ComparisonTime: 176cycl)

$$\text{Sampling Time} = 128 \times 50ns = 6.4\mu s \qquad \text{Comparison Time} = 176 \times 50ns = 8.8\mu s$$

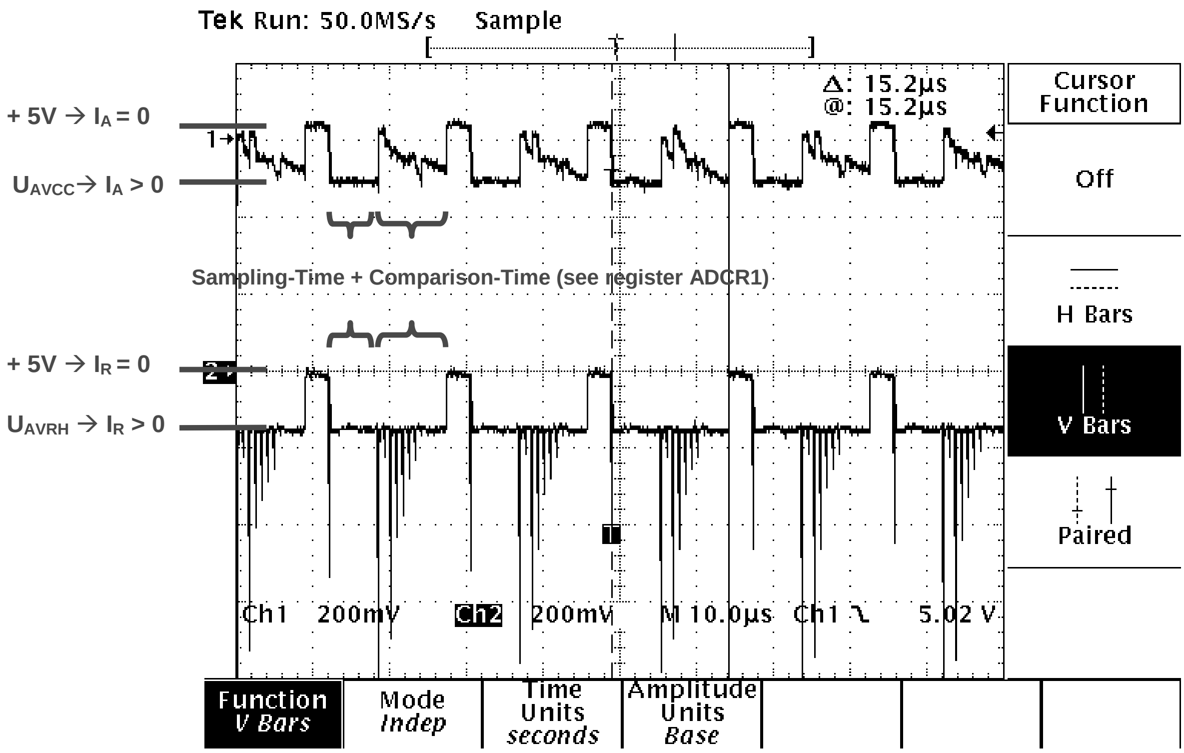


Figure 7-2: Example of measuring (Stop conversion mode)