APPLICATION NOTE

AVR1302: Using the XMEGA Analog Comparator

Atmel AVR 8-bit Microcontrollers

Features

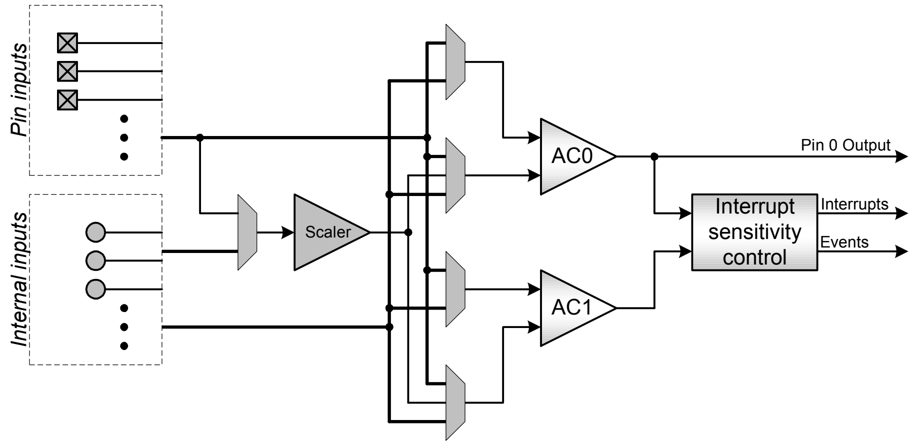
- Flexible input selection
- High-speed vs. Low-power option
- Selectable input hysteresis
- Comparator 0 Output available on I/O pin
- Scalable voltage references
- Window mode

Introduction

The XMEGA® Analog Comparator (AC) module is a high-performance dual-comparator module with flexible multiplexer (MUX) settings, integrated voltage reference scaler and several configuration options. It also has an option to combine and compare the internal comparator outputs in order to implement a window function.

This application note describes the basic functionality of the XMEGA AC with code examples to get up and running quickly. A driver interface written in C is included as well.

Figure 1. Analog Comparator Overview.



1. Module Overview

This section provides an overview of the basic configuration options and functionality of the Analog Comparator. Chapter 2 then walks you through the basic steps to get up and running, with register descriptions and configuration details.

1.1 Comparator Operation

Each comparator block has its own set of control and MUX selection registers. For all intents and purposes, they can be operated independently.

In general, if the positive input signal is above the negative input, then the comparator output is logic one, and logic zero otherwise.

Each comparator's *MUX Control* register (`ACnMUXCTRL`) selects the signals to use for the positive and negative input. The positive input can be connected to analog input pin AC0, AC1, AC2, AC4, AC5 or AC6. The negative input can be connected to analog input pin AC0, AC1, AC3, AC5 or AC7.

In addition, the inputs can also choose one of the following internal signals: The output from the 12-bit DAC, the Bandgap Reference or the internal voltage scaler for the Analog Comparator module. Only the DAC output is available on the positive input pin. The voltage scaler is covered in Section 1.5.

1.2 High-speed vs. Low-power Option

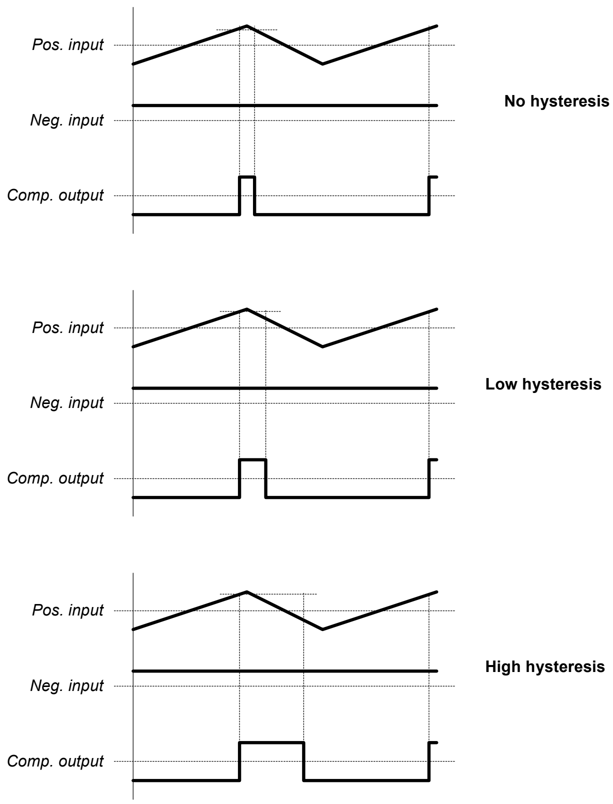
Each comparator can be configured for either high-speed performance or low-power operation. By setting the *High-speed Mode* bit (`HSMODE`) in each comparator's *Control* register (`ACnCTRL`), that comparator will be biased towards high-speed operation. Clearing the bit increases the propagation delay in the comparator but lowers the power consumption. The two comparator blocks can be configured individually.

1.3 Selectable Input Hysteresis

Each comparator can be configured for no, low or high hysteresis. For applications that compare signals that are very close to each other, increasing the hysteresis would help avoid excessive toggling of the comparator output if the signals are noisy.

The hysteresis level is configured with the *Hysteresis Mode* bit field (`HYSMODE`) in each comparator's *Control* register (`ACnCTRL`). The two comparator blocks can be configured individually. Figure 1-1 illustrates the different hysteresis settings.

Figure 1-1. Different Hysteresis Settings.



Refer to the device datasheet for details on the electrical characteristics for the hysteresis settings.

1.4 The Common Status Register

The *Status* register (`STATUS`) for the Analog Comparator module covers current status and interrupt flags for both comparator blocks.

The two *Comparator State* bits (`ACnSTATE`) are directly linked to the output of the comparator blocks and shows the current status for each comparator.

The two *Interrupt Flag* bits (`AcnIF`) get set to logic one when an interrupt condition occurs, regardless if interrupts are enabled or not. Each comparator can be configured to give an interrupt condition every time the output toggles, on positive edges on the output or on negative edges. This is configured with the *Interrupt Mode* bit field (`INTMODE`) in each comparator's *Control* register (`ACnCTRL`).

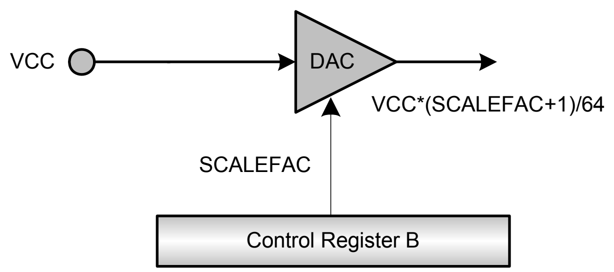
The same distinguishing between state bit and interrupt flag applies to the *Window Mode State* bit (`WSTATE`) and *Window Mode Interrupt Flag* bit (`WIF`) in the *Status* register (`STATUS`). Window mode is covered in Section 1.6.

1.5 Voltage Scaler

One of the input selections for the comparators is the internal voltage scaler connected to VCC. This is a 6-bit DAC internal in the Analog Comparator module.

Being a 6-bit DAC, voltages can be generated in 64 discrete steps from 1/64 * VCC to VCC. Figure 1-2 shows an overview of the voltage scaler.

Figure 1-2. Voltage Scaler Overview.



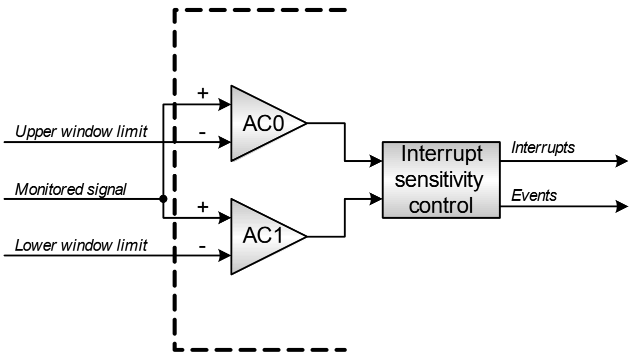
The scale factor is selected with the *Voltage Scaler Factor* bit field (`SCALEFAC`) in the *Control Register B* (`CTRLB`).

1.6 Window Mode

The window mode is intended for monitoring a signal against a certain voltage range, or window. While the comparator outputs show whether a signal is above or below another, the window mode output shows whether a signal is above, below or *inside* a voltage range.

Interrupt conditions can be given when the signal is inside, outside, above or below the voltage range. Window mode is enabled with the *Window Mode Enable* bit (`WEN`) in the *Window Mode Control* register (`WINCTRL`). The interrupt condition is configured with the *Window Mode Interrupt Mode* bit field (`WINTMODE`) in the same register.

Figure 1-3. Connection Setup for Window Mode.



Positive and negative inputs for the two comparators are selected by using the multiplexers in the same way as for traditional operation. In order to use the window function, you need to externally connect comparator 0's positive input to comparator 1's positive input, or direct both MUXes to the same input. This junction also connects to the monitored signal.

Current window state and associated interrupt flag can be found in the *Status* register (`STATUS`), as described in Section 1.4.

1.7 Comparator 0 Output Available on Output Pin

By setting the *Comparator 0 Output Enable* bit (`AC0OUT`) in *Control Register A* (`CTRLA`), the output from comparator block 0 is connector to the analog output pin AC0OUT. Comparator 0 must be configured and enabled in order for this feature to work. The output is digital, and reads logic one if the positive input is above the negative input, and vice versa.

2. Getting Started

The necessary few steps to get up and running with comparators is as follows:

1. Configure MUX settings for the comparators you want to use.
2. Configure which interrupt condition to use for each comparator.
3. Configure interrupt levels if you want to use interrupt handlers.
4. Enable the comparators you want to use.

For a detailed example comparator setup and use and how to use the window mode, study the example software for this application note.

3. Driver Implementation

This application note includes a source code package with a basic AC driver implemented in C. It is written for the IAR Embedded Workbench® compiler.

Note: This AC driver is not intended for use with high-performance code. It is designed as a library to get started with the AC. For timing and code space critical application development, you should access the AC registers directly. Refer to the driver source code and device datasheet for more details.

3.1 Files

The source code package consists of three files:

- *ac_driver.c* – AC driver source file
- *ac_driver.h* – AC driver header file
- *AC_example_polled.c* – Example code using the AC driver using one comparator with hysteresis and polled functions.
- *AC_example_window_and_interrupt.c* – Example code using the AC in window mode and with interrupts.

For a complete overview of the available driver interface functions and their use, refer to the source code documentation.

3.2 Doxygen Documentation

All source code is prepared for automatic documentation generation using Doxygen. Doxygen is a tool for generating documentation from source code by analyzing the source code and using special keywords. For more details about Doxygen, visit http://www.doxygen.org. Precompiled Doxygen documentation is also supplied with the source code accompanying this application note, available from the *readme.html* file in the source code folder.

4. Revision History

Doc. Rev.	Date	Comments
8041C	07/2013	A bug (AN-3842) is fixed
8041B	04/2008	Updated F_CPU define and GCC compatible
8041A	02/2008	Initial document release

Atmel | Enabling Unlimited Possibilities®

Atmel Corporation
1600 Technology Drive
San Jose, CA 95110
USA
Tel: (+1)(408) 441-0311
Fax: (+1)(408) 487-2600
www.atmel.com

Atmel Asia Limited
Unit 01-5 & 16, 19F
BEA Tower, Millennium City 5
418 Kwun Tong Road
Kwun Tong, Kowloon
HONG KONG
Tel: (+852) 2245-6100
Fax: (+852) 2722-1369

Atmel Munich GmbH
Business Campus
Parkring 4
D-85748 Garching b. Munich
GERMANY
Tel: (+49) 89-31970-0
Fax: (+49) 89-3194621

Atmel Japan G.K.
16F Shin-Osaki Kangyo Building
1-6-4 Osaki, Shinagawa-ku
Tokyo 141-0032
JAPAN
Tel: (+81)(3) 6417-0300
Fax: (+81)(3) 6417-0370