AN-897
APPLICATION NOTE

One Technology Way • P.O. Box 9106 • Norwood, MA 02062-9106, U.S.A. • Tel: 781.329.4700 • Fax: 781.461.3113 • www.analog.com

ADC Readback from Super Sequencers

by Enrico Del Mastro and Michael Bradley

INTRODUCTION

The ADM1062, ADM1063, ADM1064, ADM1066, ADM1069, ADM1166, and ADM1169 family of fully programmable supply sequencers and supervisors can be used as complete supply management solutions in systems with multiple voltage supplies.

These devices all feature an on-chip, 12-bit ADC. The ADCs can be set up to read a single result or to continuously read the selected channels. Averaging is provided for each channel and can be either enabled (on) or disabled (off).

This application note provides the steps to set up and read from the ADCs for a single result (with averaging on or off). It also provides the steps to read continuously from the ADCs (with averaging on or off). For the purposes of this application note, the VH channel is used as an example in the sets of instructions provided.

For more information on the features and functions of the ADM1062, ADM1063, ADM1064, ADM1066, ADM1069, ADM1166, and ADM1169 devices, refer to the relevant data sheet and to the Application Note AN-698 and the Application Note AN-721 for details on the configuration registers.

TABLE OF CONTENTS

REVISION HISTORY

8/14—Rev. 0 to Rev. A

Added ADM1166 and ADM1169 ..... Universal
Changed RRCTL to RRCTRL ..... Throughout
Changes to Title, Author, and Introduction 1
Added Guidelines for Enabling the ADC Round Robin Section, Device Specific Considerations Section, All Zeros Value Readback Section, and Static Values Readback Section 3

6/07—Revision 0: Initial Version

GUIDELINES FOR ENABLING THE ADC ROUND ROBIN

There are three ways the ADC round robin (RR) can be enabled in a Super Sequencer®.

- The go bit in the RRCTRL register (Register 0x82[0]) is set to 1 to trigger a single cycle of RR.
- The enable bit in the RRCTRL register (Register 0x82[1]) is set to 1 for continuous RR operation.
- The enable bit in the RRCTRL register (Register 0x82[1]) is set in a sequence engine state.

It is possible to use more than one of these methods simultaneously to enable the ADC RR. However, it is generally recommended to enable the ADC RR with only one of these methods at a time. The three enables are combined in a logical OR manner; therefore, if the ADC RR is enabled in the active sequence engine state, setting or clearing the enable bit (Register 0x82[1]) has no effect on the state of the ADC RR.

By using only one method to enable the ADC RR, it is clear to the user how sampling can be started and stopped. For certain Super Sequencer devices, it is also advisable to use only a single method of enabling the ADC RR to ensure correct operation, as described in the Device Specific Considerations section.

DEVICE SPECIFIC CONSIDERATIONS

Note that these device specific considerations only apply to the ADM1062, ADM1063, ADM1064, ADM1066, and ADM1069. They do not apply to the ADM1166 or ADM1169.

During normal operation of these devices, the following behaviors may be observed under certain conditions. The frequency of occurrence of these behaviors is relatively uncommon, but depends on the number and frequency of I^2C bus transactions.

All Zeros Value Readback

When reading back values from the ADC RR for the different channels, software may report a value of zero or 0x0000 for all channels from a single ADC RR iteration.

When this readback occurs, software must either wait for the next ADC RR cycle to complete, if running continuously, or trigger another single shot ADC RR cycle, as required. The next ADC RR cycle produces a fresh set of values that can be used.

It is important to note that only the readback of the ADC RR channel values over the I^2C bus is affected. The internal ADC RR registers used for warnings and directing the sequence engine are unaffected.

Static Values Readback

In certain configurations, the ADC RR output registers may report static or unchanging values even though the input supplies are changing. In this case, the ADC RR has stopped cycling and must be stopped and restarted.

Depending on the nature of the system and how the ADC RR is being enabled, it may be possible to stop and to restart the ADC RR automatically as part of the normal sequence engine operation, or the user can use an external processor to implement a software workaround.

If the ADC RR is enabled as part of a sequence engine state (for example, as part of the power-good state), static value readback can be avoided by modifying the sequence engine configuration slightly. Instead of enabling the ADC RR with a single power-good state, create a second power-good state that is identical to the original state. In the second state, disable the ADC RR. Use a long (400 ms) timeout condition to transition from the original power-good state to the new power-good state, and use a short (0.1 ms) timeout condition to jump back to the original power-good state. In this way, the ADC RR is continuously enabled and disabled, and the ADC values read are never static.

If using single shot mode with the ADC RR, check the go bit (Register 0x82[0]) after every single shot to see if the bit is 1. If it is, the bit must be written to 0.

If using continuous ADC RR mode, the software must detect if the readback values are static and stop and restart the ADC RR. This is done by setting the enable bit (Register 0x82[1]) to 0 and back to 1.

If more than one method is simultaneously used to enable the ADC RR, and the output values are static, all the enable methods must be disabled to allow the ADC RR to stop and restart.

SINGLE READ (AVERAGING OFF)

1. Set up Register 0x80 and Register 0x81 (RRSEL1 and RRSEL2, respectively). These registers select the channels that are monitored.
   0 = channel selected
   1 = channel not selected
   For example, to select the VH channel, write 0xEF to Register 0x80 and 0x1F to Register 0x81
2. Set the go bit by writing 0x01 to Register 0x82 (RRCTRL).
   Go = 1
   Enable = 0
   Average = 0
   STOPWRITE = 0
   CLEARLIM = 0
3. Enter a loop with a timeout and continuously read the data from Register 0x82 (RRCTRL).
   - If Register 0x82 = 0x00, exit the loop (the go bit is reset).
   - If Register 0x82 ≠ 0x00, continue around the loop.
4. If the go bit is reset, write 0x08 to Register 0x82 (RRCTRL).
   Go = 0
   Enable = 0
   Average = 0
   STOPWRITE = 1
   CLEARLIM = 0
5. Read the registers associated with VH.
   0xA8 (ADCHVH)
   0xA9 (ADCLVH)
6. Reset the STOPWRITE bit. Write 0x00 to Register 0x82.
   Go = 0
   Enable = 0
   Average = 0
   STOPWRITE = 0
   CLEARLIM = 0

SINGLE READ (AVERAGING ON)

1. Set up Register 0x80 and Register 0x81 (RRSEL1 and RRSEL2, respectively). These registers select the channels that are monitored.
   0 = Channel selected
   1 = Channel not selected
   For example, to select the VH channel, write 0xEF to Register 0x80 and 0x1F to Register 0x81.
2. Set Register 0x82 (RRCTRL) to 0x05.
   Go = 1
   Enable = 0
   Average = 1
   STOPWRITE = 0
   CLEARLIM = 0
3. Enter a loop with a timeout and continuously read the data from Register 0x82 (RRCTRL).
   - If Register 0x82 = 0x04, exit the loop (the go bit is reset).
   - If Register 0x82 ≠ 0x04, continue around the loop.
4. If the go bit is reset, write 0x0C to Register 0x82 (RRCTRL).
   Go = 0
   Enable = 0
   Average = 1
   STOPWRITE = 1
   CLEARLIM = 0
5. Read the registers associated with VH.
   0xA8 (ADCHVH)
   0xA9 (ADCLVH)
6. Reset the STOPWRITE bit. Write 0x04 to Register 0x82.
   Go = 0
   Enable = 0
   Average = 1
   STOPWRITE = 0
   CLEARLIM = 0

CONTINUOUS READ (AVERAGING OFF)

1. Set up Register 0x80 and Register 0x81 (RRSEL1 and RRSEL2, respectively). These registers select the channels that are monitored.
   0 = channel selected
   1 = channel not selected
   For example, to select the VH channel, write 0xEF to Register 0x80 and 0x1F to Register 0x81
2. Write 0x02 to Register 0x82 (RRCTRL).
   Go = 0
   Enable = 1
   Average = 0
   STOPWRITE = 0
   CLEARLIM = 0
3. Write 0x0A to Register 0x82 (RRCTRL).
   Go = 0
   Enable = 1
   Average = 0
   STOPWRITE = 1
   CLEARLIM = 0
4. Read the registers associated with VH.
   Register 0xA8 (ADCHVH)
   Register 0xA9 (ADCLVH)
5. Write 0x02 to Register 0x82 (RRCTRL).
   Go = 0
   Enable = 1
   Average = 0
   STOPWRITE = 0
   CLEARLIM = 0

CONTINUOUS READ (AVERAGING ON)

1. Setup Register 0x80 and Register 0x81 (RRSEL1 and RRSEL2, respectively). These registers select the channels that are monitored.
   0 = channel selected
   1 = channel not selected
   For example, to select the VH channel, write 0xEF to Register 0x80 and 0x1F to Register 0x81.
2. Write 0x06 to Register 0x82 (RRCTRL).
   Go = 0
   Enable = 1
   Average = 1
   STOPWRITE = 0
   CLEARLIM = 0
3. Write 0x0E to Register 0x82 (RRCTRL).
   Go = 0
   Enable = 1
   Average = 1
   STOPWRITE = 1
   CLEARLIM = 0
4. Read the registers associated with VH.
   0xA8 (ADCHVH)
   0xA9 (ADCLVH)
5. Write 0x06 to Register 0x82 (RRCTRL).
   Go = 0
   Enable = 1
   Average = 1
   STOPWRITE = 0
   CLEARLIM = 0

I^2C refers to a communications protocol originally developed by Philips Semiconductors (now NXP Semiconductors).