TMS470R1x Multi-Buffered Analog-to-Digital Converter (MibADC) Reference Guide

Literature Number: SPNU206B
September 2005

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and application s using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation.

Resale of TI products or services with statements different form or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

Products		**Applications**	
Amplifiers	amplifier.ti.com	Audio	www.ti.com/audio
Data Converters	dataconverter.ti.com	Automotive	www.ti.com/automotive
DSP	dsp.ti.com	Broadband	www.ti.com/broadband
Interface	interface.ti.com	Digital Control	www.ti.com/digitalcontrol
Logic	logic.ti.com	Military	www.ti.com/military
Power Mgmt	power.ti.com	Optical Networking	www.ti.com/opticalnetwork
Microcontrollers	microcontroller.ti.com	Security	www.ti.com/security
		Telephony	www.ti.com/telephony
		Video & Imaging	www.ti.com/video
		Wireless	www.ti.com/wireless

Mailing Address: Texas Instruments
Post Office Box 655303 Dallas, Texas 75265

REVISION HISTORY

REVISION	DATE	NOTES
B	9/05	Table 1: Added ADIN to formula in Results Row, changed 1023 to 1024. Figure 6: Removed the extra "10-bit 2us analog-to-dig" from the block diagram. Page 10: Changed 1023 to 1024 in Eq. 1. Added statement “where . . . “ after equation. Figure 11: Added "ADC Core Timing" diagram. Page 28: Changed completely the self-test mode and calibration mode descriptions. Table 9: Changed second column heading. Section 11.4: Rewrote introductory paragraphs to ADEISR register. Section 11.23: Section 11.24, changed register names. Section 11.5, Section 11.6, and Section 11.7: Changed “MibADC” to “ADC” in first paragraph. Section 13: Changed the Analog Input model.
A	3/04	Updates: Section 11.12: Updated Bit 4 bit definition. Section 11.12: Added note concerning interrupt handling procedure.
*	12/02	Initial version

Contents

Figures

Tables

Multi-Buffered Analog To Digital Converter (MibADC)

This reference guide discusses the features, operation, specifications, and initialization of the TMS470R1VX 16-channel, multibuffered, 10-bit analog-to-digital converter (MibADC). The analog converter uses 10-bit sampling, successive approximation register-based architecture. The upper four bits are converted using a switched capacitor charge redistribution technique; the lower six bits are derived from a voltage-scaled reference (resistor string between AD_{REFHI} and AD_{REFLO}). The buffered MibADC functions in compatibility mode or in buffered mode.

Topic **Page**

1 Overview

This section provides an overview of the MibADC module. Table 1 contains a brief description of the MibADC, lists its significant pins, describes the priority scheme and interrupts, how conversions are triggered, and the expected results of the conversion.

Table 1. MibADC Module Overview

Description	The analog-to-digital converter accepts an analog signal and converts it to a 10-bit digital value. Typical uses include converting a sensor voltage to a digital value for digital manipulation.
Pins	**V_{CCAD}**: Analog-to-digital converter power **V_{SSAD}**: Analog-to-digital converter ground **AD_{REFHI}**: Analog-to-digital converter high reference. Any signal with a voltage higher than AD_{REFHI} is recorded as 0x3FF. $AD_{REFHI} \leq V_{CCAD.}$ **AD_{REFLO}**: Analog-to-digital converter low reference. Any signal with a voltage lower than AD_{REFLO} is recorded as 0x00. $AD_{REFLO} \geq V_{SSAD.}$ **ADIN[15:0]:** Analog-to-digital converter input channels. (A device can have up to 16 input channels.) **ADEVT**: Analog-to-digital event trigger. Signals on this line can trigger a group conversion. **Internal Event Triggers:** In addition to the ADEVT pin, the MibADC includes three additional event triggers from internal sources, for example, internal timers.
Priority	Within a group, the lowest channel has priority over higher channels. The group priority is event group (highest), group 1, and group 2 (lowest); this group priority takes effect when groups are started simultaneously. When a second group is triggered while another group is being converted, the group being converted finishes conversion and then the second group begins its conversion; if the first group is freeze enabled, the second group interrupts the conversion of the first group.
Programmer's Models	**Compatibility Mode:** Programmer's model is compatible with the TMS470R1X ADC module. Conversion results are stored in digital result registers. **Buffered Mode:** Digital result registers are replaced with three FIFO buffers, one for each conversion group. Buffers can be serviced by interrupts or by DMA.
Operating Modes	**Conversion Mode:** Normal active mode for converting analog input pins **Calibration Mode:** Active mode for calibrating out the midpoint offset error of the MibADC. **Self-test Mode:** Active mode for detection of faults on the analog input pins. **Power-down Mode:** Inactive mode for low power consumption.
Interrupts	Interrupt request of each group can be individually enabled. Buffered mode provides an interrupt threshold counter for FIFO buffer of each group.
DMA Capability	Three DMA channels. The MibADC can be serviced by DMA in buffered mode only.

Table 1. MibADC Module Overview (Continued)

Begin Conversion	For the event group, the external trigger on the ADEVT pin triggers the conversion (after ADEISR is initialized). Optionally, events can be triggered from an internal timer. For groups 1 and 2, writing to the ADISR1 or ADISR2 (other than 0x00) initiates the group conversion. Group 1 can also be event triggered.	
Results	Input Voltage on the ADINx Pin	Value in the Digital Result Register (DRR)
	Input voltage between maximum recommended voltage and AD_{REFHI}	0x3FF
	Input voltage for $AD_{REFHI} > ADIN \geq AD_{REFLO}$.	$\approx \frac{1024 \times (Input\_Voltage - AD_{REFLO})}{(AD_{REFHI} - AD_{REFLO})}$
	Input voltage between AD_{REFLO} and minimum recommended voltage	0x00

1.1 Definitions

The following define some of the important terminology used in this document.

- Sample and hold—Internal capacitor array that averages the signal over the acquisition time.
- AD clock period:
  $t_{c(ADCLK)} = t_{c(ICLK)}$*PS.2:0

1.2 Features

The MibADC module converts analog voltages to 10-bit digital values; the module includes internal sample-and-hold circuitry capable of sampling one of the sixteen analog inputs at a time. A sequencer controls the order and the timing of conversions, with input channels organized in three conversion groups. Each input channel may be assigned to one or more of the conversion groups.

The conversion groups may be configured for either in single or continuous conversion modes. Each of the conversion groups may be triggered by software, and two of the three conversion groups may be event triggered. The

event is selectable for each group from one of four sources; three internal sources (from timer modules on the same device) and one MibADC controlled pin (ADEVT).

The result of conversions are stored in either in individual digital result registers (compatibility mode) or in three FIFOs (buffered mode) where each FIFO holds the results from one of the conversion groups.

The module supports interrupts in a flexible manner as described in section 8, *Interrupt Generation*. In addition, the module supports DMA accesses in buffered mode.

The main features of the MibADC include:

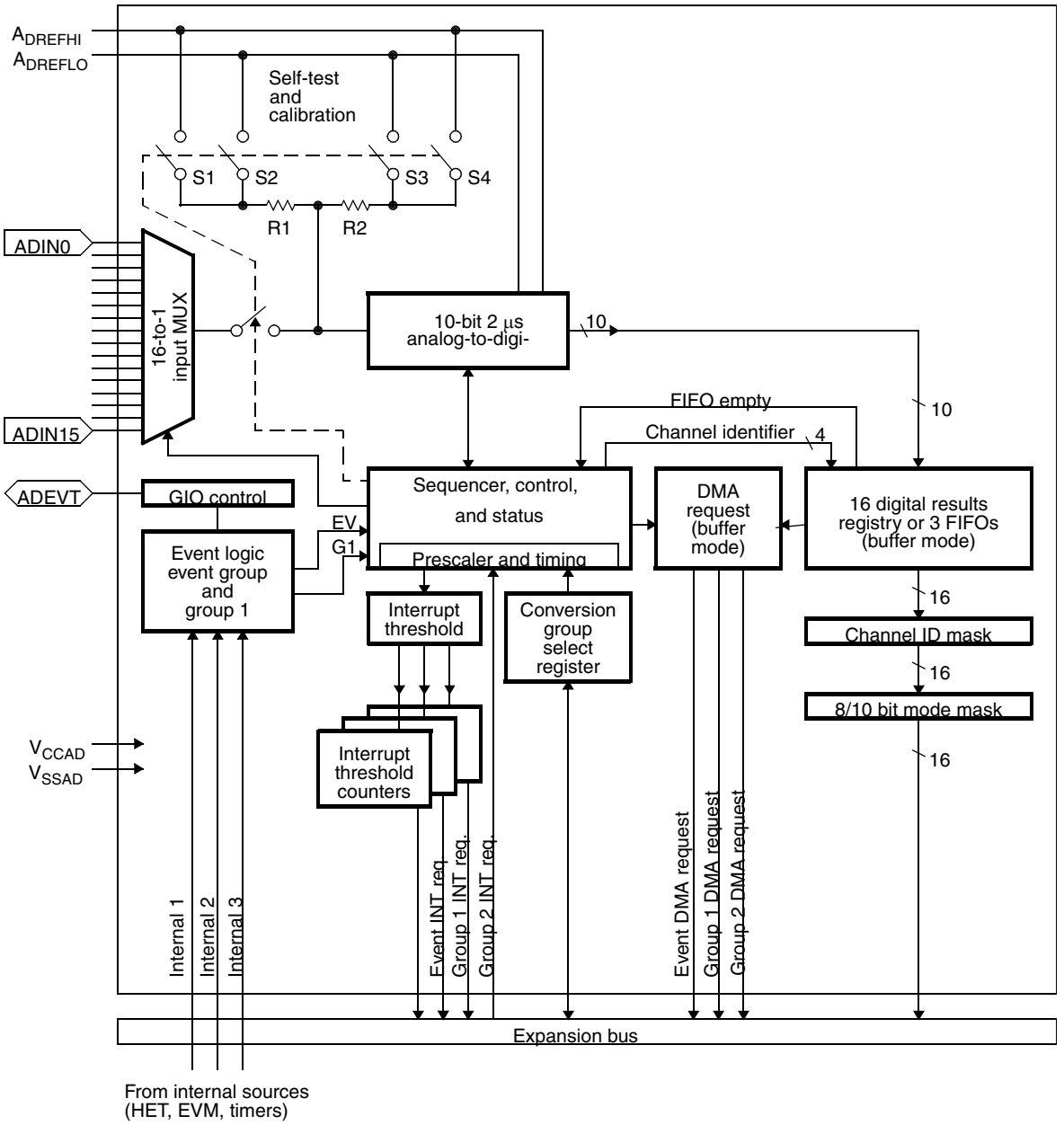
- 10-bit resolution
- AD_{REFHI} and AD_{REFLO} pins (high and low reference voltages)
- Total Sample/Hold/Convert time: 1.75 μs Typical Minimum (see datasheet)
- Two programmer's models (select one or the other)
  - Unbuffered (compatibility):
    - The programmer's model matches the TMS470R1x ADC module.
    - One digital result register is available for each of the 16 MibADC channels.
    - Results are stored in two-port RAM (some RAM unused).
    - Data is organized by channel number.
  - Buffered:
    - The programmer's model is MibADC with three FIFO buffers.
    - One FIFO buffer per conversion group is available (event, group 1, group 2).
    - The size of FIFO buffers is adjustable because allocation of channels to conversion groups is completely programmable.
    - The FIFO is serviced either by interrupt or by DMA.
    - A programmable interrupt threshold counter is available for each FIFO.
    - An option is available to store channel ID with data in FIFO mode for enhanced robustness.
    - An option is available to read from FIFO as either 8-bit or 10-bit value. Saves shift operations for software using only the eight MSBs of the conversion result.

- ❏ There are sequential multichannel conversions (up to 16 channels in ascending order).
- ❏ Single or continuous conversion modes are available.
- ❏ One or two software-controlled conversions are available: up to 16 channels converted on a software request.
- ❏ One or two event-initiated conversions are available: up to 16 channels converted on an external event or internal timer event.
- ❏ The MibADC is interrupt driven, DMA driven (buffered mode only), or polled operation (end-of-conversion sequence).
- ❏ A prrogrammable MibADC clock prescaler (ADCLK) to optimize conversion rate is available.
- ❏ Each conversion group has a programmable acquisition time:
  - If SEN (ADSAMPEV.15 is 0, then for compatibility with ADC, the MibADC sample time is determined by ACQ.1:0 (ADCR1.4:3)
  - If SEN (ADSAMPEV.15) is set to 1, then the sample time for each group is programmable and controlled by EVACQ.7:0 (ADSAMPEV.7:0) for the event group, G1ACQ.7:0 for group 1, and G2ACQ.7:0 (ADSAMP2.7:0) for group 2.
- ❏ The MibADC has an embedded self-test.
- ❏ The MibADC has embedded calibration logic.
- ❏ Power-down mode can be used.
- ❏ The external event pin (ADEVT) is programmable as general-purpose I/O.
- ❏ Three device-specific internal timer events are available.

2 Functional Description

This section presents a general discussion of the components and operation of the TMS470 10-bit MibADC. Figure 1 illustrates the components of the MibADC module.

Figure 1. MibADC Block Diagram



2.1 Detailed MibADC Diagrams

The expanded MibADC diagram in Figure 2 shows compatibility mode in more detail; Figure 3 shows buffered mode in more detail. Inside the group status and control logic block is a matrix of function versus group register bits. As an example, the STOP bit of group 2 is in the ADSR register, bit 9.

Figure 2. MibADC Compatibility Mode Functional Block Diagram

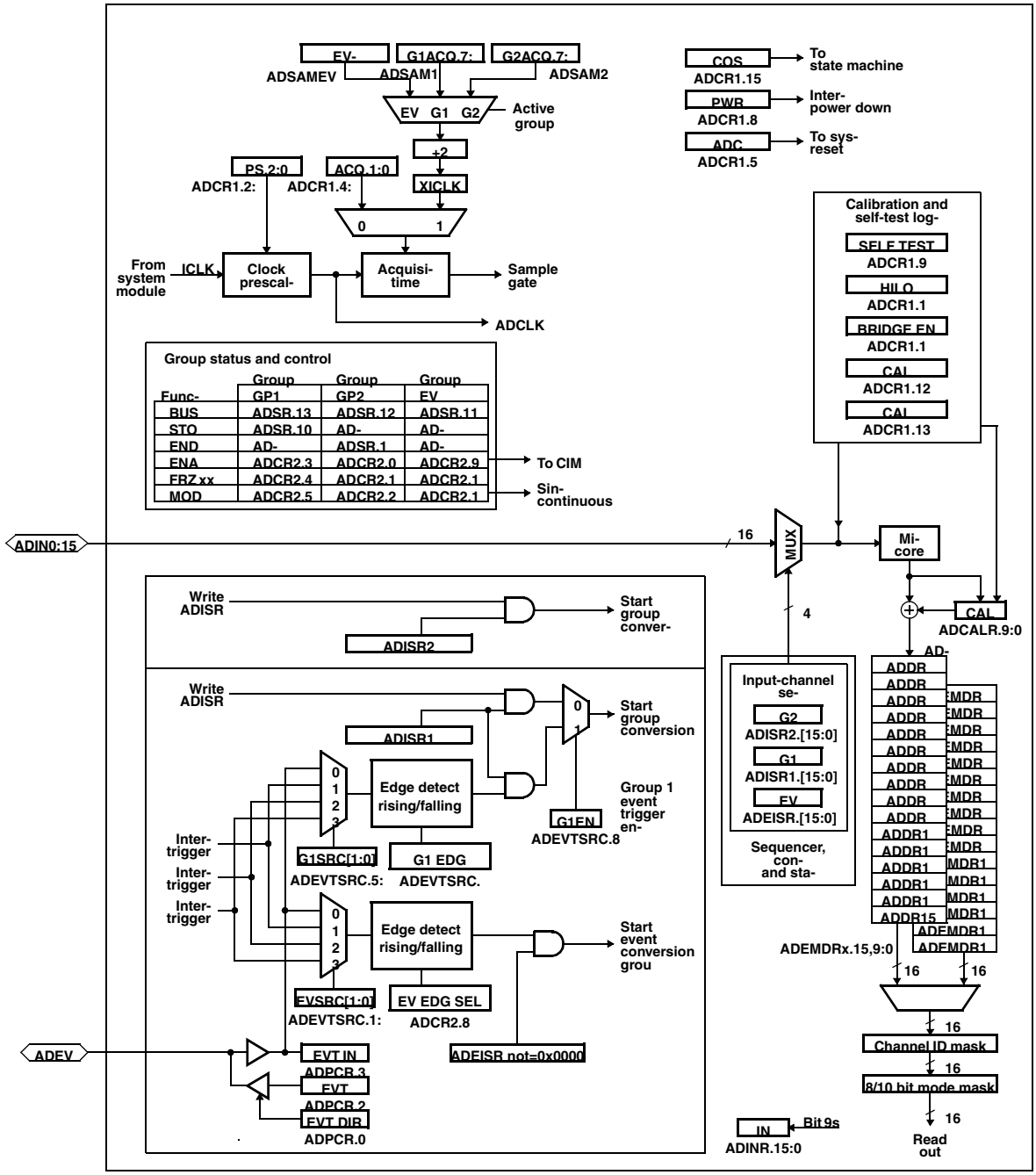
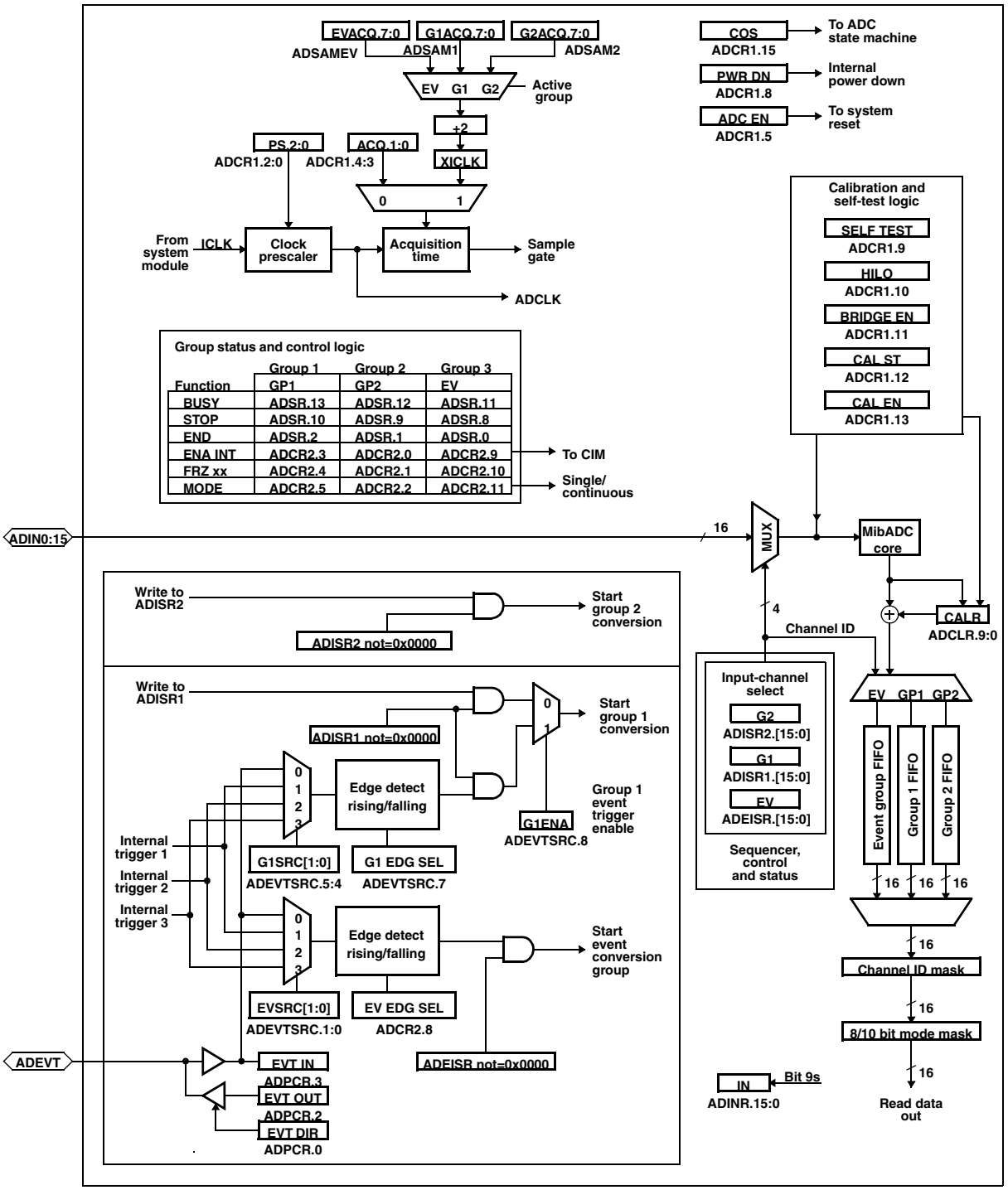


Figure 3. MibADC Buffered Mode Functional Block Diagram



2.2 Functional Blocks

This section contains a functional description of each block in the MibADC (see Figure 1 and Figure 3).

2.2.1 Input Multiplexer

The input multiplexer (MUX) connects the selected input channels (pins ADIN[15:0]) to the input V_{in} of the analog converter core (Figure 1). The selected input channel is determined by the conversion group that has priority. The channels chosen for conversion are programmed into one or more of the three conversion group selection registers: group 1 input-select register (ADISR1), group 2 input-select register (ADISR2), or the event group input-select register (ADEISR).

2.2.2 Analog-to-Digital Converter Core

The MibADC core has a separate power bus (V_{CCAD} and V_{SSAD}) for its analog circuitry. This power bus enhances MibADC performance by reducing digital switching noise present on the digital power lines (V_{CC} and V_{SS}), which can couple into the MibADC analog stage.

The converter core uses a 10-bit sampling, successive approximation type architecture. The analog conversion range is determined by the reference voltages: AD_{REFHI} (high reference voltage) and AD_{REFLO} (low reference voltage). These voltages are, respectively, the maximum and the minimum voltages that can be converted. Both AD_{REFHI} and AD_{REFLO} must be chosen not to exceed the analog power supplies (V_{CCAD} and V_{SSAD}).

Analog input voltages greater than AD_{REFHI} are converted to 0x3FF; input voltages less than AD_{REFLO} are converted to 0x000. Voltages between AD_{REFHI} and AD_{REFLO} produce a conversion result that is proportional to the difference of [AD_{REFHI} − AD_{REFLO}]. The digital result of the conversion process is approximated by Equation 9:

$$DigitalResult = \frac{1024 x InputVoltage - AD_{REFLO}}{(AD_{REFHI} - AD_{REFLO})} \qquad \textit{(EQ 1)}$$

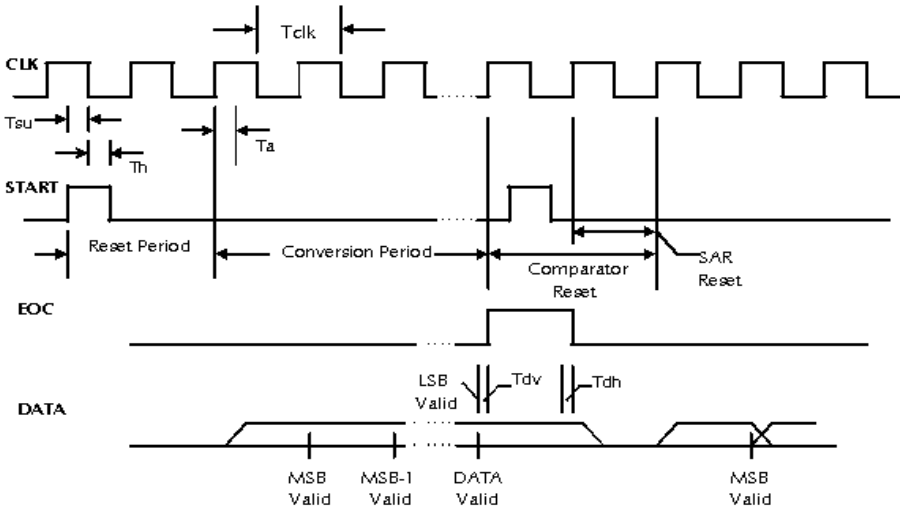
where $AD_{REFHI} > ADIN \geq AD_{REFLO}$. If $ADIN \geq AD_{REFHI}$, the conversion result will be 0x3FF.

The MibADC acquisition period starts when either ADISR1 or ADISR2 is written with non-zero data, or when an event trigger initiates an event-triggered group. The acquisition bits should be programmed to

guarantee that the sample-and-hold circuitry has time to adequately sample the input voltage (see section 12.1, *Sample/Hold Time*). During this acquisition time, the converter stage is maintained in its reset state. The input signal is sampled directly onto the switched capacitor array, which provides an inherent sample-and-hold function.

When the acquisition time has elapsed, the converter reset is released, which allows the conversion of the sampled voltage to start. The ADC core is synchronized with the internal ADC clock (ADCLK). See section 12.2, *Total Conversion Time*. The digital result is determined one bit at a time, starting with the most significant bit (MSB). When all channels within a group have been converted, the group end (xx END) flag is set, and the digital result is then available in the corresponding digital result register.

Figure 4. ADC Core Timing



2.2.3 Conversion Group Selection Registers

The MibADC module can preset three groups of channels to be converted (always from lowest selected channel to highest). The input-select register 1 (ADISR1), input-select register 2 (ADISR2), and the event input-select register (ADEISR) define the channels associated with a conversion group. Any channel can be selected in any group. The same channel can be selected in more than one group.

Note: Channels in Conversion Groups

The logic circuitry does not prevent selecting the same channel in more than one group. A channel is converted for each group conversion in which it is selected. It is also possible to select all channels or no channels within a group.

The channels are fully controlled by software.

- ❏ Any write from the CPU to group 1 (ADISR1) or group 2 (ADISR2) with a value other than 0x0000 configures and starts the conversion of that group.
- ❏ Group 1 (ADISR1) can be either software-triggered or event-triggered. If group 1 is selected for event trigger, then conversion of this group begins when the event trigger occurs, not when ADISR1 is written as described above.
- ❏ When it is configured by software, the third group (EVT) is triggered by an external event at the ADEVT pin or from an internal timer.

For each group, the control register ADCR2 defines the following attributes:

- ❏ Conversion mode: single or continuous (See Section 5.1, *Conversion Mode*.)
- ❏ Conversion arbitration level (finishes or freezes the current conversion sequence when another group conversion starts) (See Section 6, *Conversion Priority and the Freeze-Enable Bit*.)
- ❏ Interrupts to the CPU (See Section 8, *Interrupt Generation*.)

Note: Continuous Conversion for Multiple Groups

The software-controlled group 1 (G1), group 2 (G2), and the event-triggered group cannot operate together in continuous conversion mode. When this situation occurs, group 2 is automatically reset to single conversion mode, and the G2 MODE bit (ADCR2.2) is cleared to reflect the actual conversion mode of group 2.

2.2.4 Event Edge Detector

The ADEVT pin is intended to start an event group conversion from an external trigger signal; however, if the ADEVT pin is not used to launch an event-controlled conversion, it can be used as a digital input/output function through the ADPCR register. If the application needs to use the ADEVT pin as a general-purpose I/O, then it should be noted that any edge on the

ADEVT pin, even when created by software using the ADEVT pin as digital I/O, can trigger the event group.

If the application is not designed to trigger the event group when software toggles the ADEVT pin, the two options are:

- ❏ Program the ADEISR register to 0x0000 to disable the event group.
- ❏ Program the event group for one of the other event sources in the ADEVTSRC register.

When the ADEISR register is not 0x0000 and the event that is selected by EVSRC.1:0 (ADEVTSRC.1:0) occurs, all channels selected in the event group register (ADEISR) are sequentially converted (from lowest to highest). The edge that starts a conversion can be selected as a low-to-high or a high-to-low transition; this selection is configured through the event edge select bit, EV EDG SEL (ADCR2.8).

When group 1 is selected for event trigger by G1ENA (ADEVTSRC.8), conversion is similar to that of the event group described above. The event source for group 1 is selected by G1SRC.1:0 (ADEVTSRC.5:4) and the edge polarity for this event is selected by G1EDGESEL (ADEVTSRC.7).

Tip: Software Trigger for Event Group

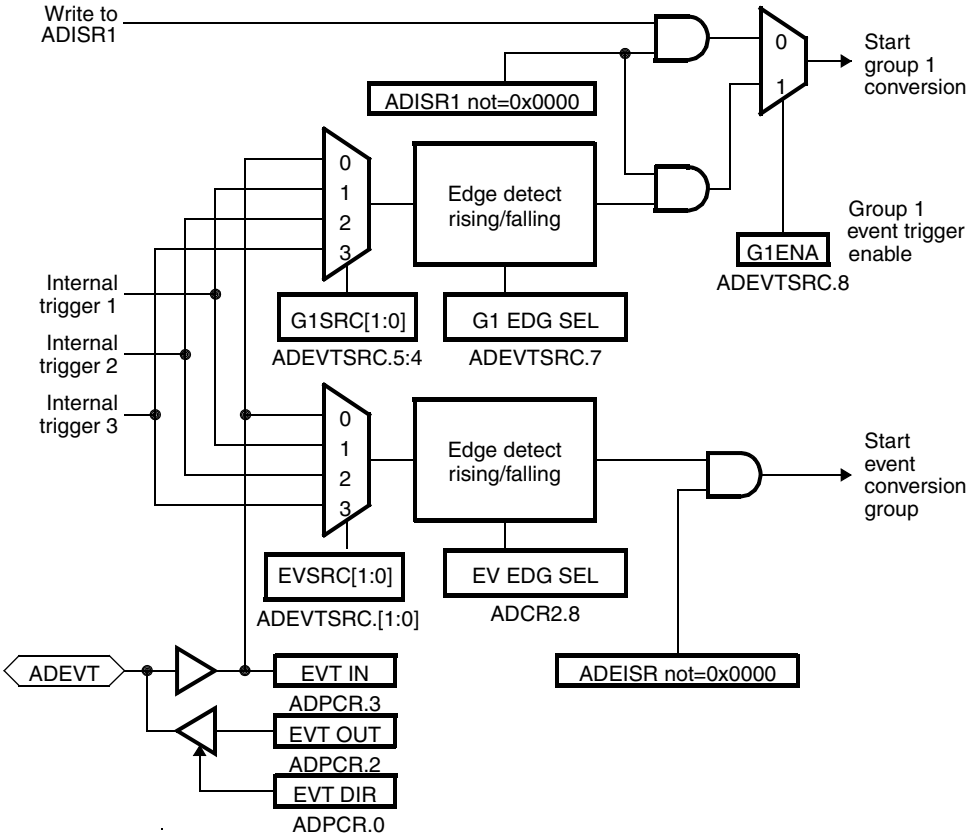
The ADEVT pin, configured as an output by setting EVT DIR (ADPCR.0) to 1, can be used to initiate a conversion of the event group by toggling the EVT OUT bit (ADPCR.2) to create the appropriate edge at the ADEVT pin. See Figure 5.

On reset, the pin is initialized as input (EVT DIR = 0), and the ADEISR register is 0x0000.

2.2.5 Event Source Selection

A new feature of the TMS470R1VX buffered MibADC is that the event group and group 1 can be programmed to trigger a conversion based upon one of these internal event sources in addition to an edge on the ADEVT pin. Typically, these internal sources originate from timers on the device. (For more detailed information, see section 11.28, *AD Event Source Register (ADEVTSRC)*.)

Figure 5. ADEVT Pin Block Diagram



2.2.6 Self-Test and Calibration Functions

The MibADC includes specific hardware for detecting an ADC pin failure and supporting the application program during the calibration phase. This function is initiated when self-test mode or calibration operations are enabled through the ADCR1 control register. See section 5.2, *Self-Test Mode* for additional information.

2.2.7 Offset Error Correction

The offset error correction block adjusts the digital result output from the ADC core before storing the data in the digital result register frame. See section 5.3, *Calibration Mode*, for additional information.

A dedicated 10-bit read/write register ADCALR (calibration and correction register) collects the calibration data and stores the offset error correction value generated by user software. During calibration operations, the CPU reads the ADCALR register to acquire the calibration data and compute an

offset error correction value. At the end of calibration operations, the CPU writes the computed offset error correction value to the ADCALR register.

The ADCALR register and an adder provide an offset cancellation mechanism. This automatic correction is enabled in noncalibration mode only. The digital result output from the ADC core is added to the contents of the ADCALR register before it is stored in the digital result register frame in compatibility mode, or before it is stored in the FIFO in buffered mode.

Note: Store Error Correction Value in Correct Form.

Always store the calculated offset error correction value in ADCALR in 2s complement form.

2.2.8 Sequencer, Control, and Status Block

The sequencer, control, and status block coordinates the operations of the MibADC, including the input multiplexer, the converter, and the result registers. In addition, the logic of the sequencer sets the status register flags when the conversion is ongoing (G1 BUSY, G2 BUSY, or EV BUSY), stopped (G1 STOP, G2 STOP, or EV STOP), or finished (G1 END, G2 END, or EV END).

The logic controls handshaking within the three conversion groups and handles CPU read/write operations, interrupts, halts, resets, and addressing within the module. The logic in this block also supports self-test mode and calibration operations. This logic block generates the internal ADC clock that controls ADC timing.

In buffered mode, this block is responsible for buffer management, overflow detection, and generation of DMA requests.

2.2.9 Compatibility Mode Prescaler and ADC Timing

The MibADC applies a prescaler value, PS.2:0 (ADCR1.2:0) to ICLK to derive an ADC clock (ADCLK) period, $t_{c(ADCLK)}$. The total time required for acquisition and conversion is a function of ICLK, the prescaler value, and the acquisition period selections.

An acquisition prescaler selection, ACQ.1:0 (ADCR1.4:3) is provided so the MibADC acquisition time can be increased for sources with greater than 10-ohm source impedance. After the acquisition time, changing the analog input does not affect the conversion result.

See section 12.1, *Sample/Hold Time*, and section 12.2, *Total Conversion Time*.

Note: ADCLK Period Parameters

The electrical specifications and timing requirements for the analog core on TMS470R1Vx devices are different than those on TMS470R1x devices because of the different process used. Always check the electrical specifications in your device's data sheet to verify that your application meets the MibADC specifications.

2.2.10 MibADC Enhancements to Prescaler and ADC Timing

The MibADC provides the option to program the sample time for each conversion group: ADSAMPEV, ADSAMPG1, and ADSAMPG2. For compatibility with ADC, the MibADC defaults to use ACQ.1:0 (ADCR1.4:3) to determine the sample time for all three groups. To make the MibADC use the new register to control sample time, the SEN bit (ADSAMPEV.15) must be set. When this bit is set, ACQ.1:0 (ADCR1.4:3) is ignored and the extension registers ADSAMPEV, ADSAMPG1, and ADSAMPG2 control the sample time.

3 General Operation of the MibADC

The MibADC is relatively autonomous after it is correctly initialized (see Figure 1). The internal clock frequency and acquisition time must be set, the channels to be converted must be stored in at least one group select register, and as an option, the interrupt control bits can be programmed. The conversion can then be triggered by hardware or software, and the data from each converted channel can be read by the CPU after the interrupt signals an end-of-conversion. In addition to this simple operation, the hardware provides for more complex operations:

- ❏ Real-time failure detection
- ❏ Calibration through sequences implemented in software
- ❏ Self-offset error correction

The two operational modes (single or continuous conversion) associated with three groups of channel selection enable conversion of multiple combinations of sets of channels without CPU accesses. This feature reduces the amount of software overhead required.

The TMS470R1Vx peripheral clock, ICLK, is divided down by a programmable prescaler to produce ADCLK. The prescaler value controls the total conversion time.

The self-offset error corrected digital data values are stored in either the 16-bit digital results register or in three FIFOs, depending on whether the device is in compatibility mode or buffered mode.

4 Programmer Options

Two programming options are available for the MibADC module. You can select compatibility mode or buffered mode of operation.

4.1 Compatibility Mode

In compatibility mode, the MibADC programmer's model is exactly the same as the TMS470R1X ADC module. All control registers below offset 0xA0 maintain their present definition. The MibADC control registers appear to the programmer as listed in Table 2.

Note: MibADC / ADC Compatibility

Although the TMS470R1VX MibADC and TMS470R1X ADC modules share the same programmer's model and are software compatible when the MibADC is in the default compatibility mode, the electrical specifications and timing requirements of the analog core on TMS470R1Vx devices are different because of process differences. Before migrating from the TMS470R1x ADC to the TMS470R1Vx MibADC, always verify by consulting the TMS470R1Vx device datasheet that the MibADC electrical specifications and timing requirements of the TMS470R1Vx MibADC are met by your application.

Note: MibADC / Compatibility Mode II

A common question that arises is whether compatibility mode features are disabled in buffered mode, or whether buffered mode features are available in compatibility mode. In general, unless otherwise specified, it is left to the software to not use compatibility mode features in buffered mode and vice-versa (rather than a hardware lock-out). For example, it is possible to enable interrupts after the end of a conversion group in buffered mode (using ADCR2). However, this is not the intended use—in buffered mode, interrupts are intended to be generated by the interrupt threshold registers (ADTHREV, ADTHRG1, ADTHRG2) and the interrupt enables in the buffer control register (ADBCR3). Again, it is left to the programmer to enable interrupts that are consistent with the mode in which the MibADC is being used.

Table 2. Compatibility Mode MibADC Control Registers

Address	Mnemonic	Name	Description
0x00	ADCR1	Control register 1	Contains control bits for initiating and controlling calibration and conversions
0x04	ADCR2	Control register 2	Contains control bits for initiating and controlling modes and events
0x08	ADSR	Status register	Contains flags showing the status of conversion activities
0x0C	ADEISR	Event group input-channel select register	Selects channels to be converted by the event group
0x10	ADISR1	Group 1 input-channel select register	Selects channels to be converted by group 1
0x14	ADISR2	Group 2 input-channel select register	Selects channels to be converted by group 2
0x18	ADCALR	Calibration and offset error correction register	Holds calibration data from ADC core
0x1C – 0x98	ADDR15-ADDR0	Digital results registers (15 through 0)	Contains status flag and digital result of associated channel
0x20 ~ 0x98	ADEMDR15-ADEMDR0	Emulation digital results registers (15 through 0)	Mirrors values in ADDR0-15 registers but is not cleared by a read
0x9C	ADINR	Data input register	Contains bit 9 of the digital results registers, ADDR[15:0], of each of the 16 channels
0xA0	ADCPCR	Pin control register	Contains digital value of current level on ADEVT pin and pin control bits
0xB0	ADSAMPEV	Sample Control Register Event	Event group sample time
0xB4	ADSAMP1	Sample Control Register 1	Group 1 sample time
0xB8	ADSAMP2	Sample Control Register 2	Group 2 sample time
0xBC	ADBCR1	Buffer Control Register 1	Buffer enable/disable, allocation
0xC0	ADBCR2	Buffer Control Register 2	Buffer end pointer/size control
0xC4	ADBCR3	Buffer Control Register 3	Buffer DMA and interrupt enable bits
0xC8	ADBST	Buffer Status Register	Buffer empty, overrun, and interrupt flags
0xCC	ADTHREV	Event Group Interrupt Threshold	Sets fill level of event group FIFO before interrupt generated
0xD0	ADTHRG1	Group 1 Interrupt Threshold	Sets fill level of group 1 FIFO before interrupt generated

Table 2. Compatibility Mode MibADC Control Registers (Continued)

Address	Mnemonic	Name	Description
0xD4	ADTHRG2	Group 2 Interrupt Threshold	Sets fill level of group 2 FIFO before interrupt generated
0xD8	ADEVTSRC	Event Source Register	Selects extended event trigger sources for ADEVT group and group 1, and enables group 1 to act as an event triggered group

4.2 Buffered Mode

You can also select the buffered mode of operation in which conversion results are stored in FIFOs instead of the results registers. Table 3 lists the buffered mode control registers.

4.2.1 Buffer Control Registers

The three buffer control registers control the operation of the device in buffered mode. They provide the following control bits:

- ❏ Control bit that switches between compatibility and buffered modes
- ❏ Boundary registers that determine the allocation of the total available FIFO memory between the three MibADC conversion groups
- ❏ Interrupt bits
- ❏ DMA control and status bits

4.2.2 Interrupt Threshold Registers

For each of the three FIFOs, the interrupt threshold (i.e., the number of sample/conversion results stored in the FIFO before an interrupt to the CPU is generated) is programmable. This feature may significantly reduce the CPU load caused by MibADC interrupts by reducing the interrupt rate.

4.3 Sample Time Registers

The programmer's model includes a separate sample time register for each MibADC conversion group. When enabled, these registers override the global sample time programmed into control register 1 (ADCR1.4-3). In compatibility mode, these registers can also be enabled to take advantage of the updated sampling time without changing the storage scheme for conversion result.

- If SEN (ADSAMPEV.15) is 0, then for compatibility with ADC, the MibADC sample window (SW) is determined by ACQ.1:0 (ADCR1.4:3).
- If SEN (ADSAMPEV.15) is set to 1, then the sample window (SW) for each group is programmable and controlled by EVACQ.7:0 (ADSAMPEV.7:0) for the event group, G1ACQ.7:0 for group 1, and G2ACQ.7:0 (ADSAMP2.7:0) for group 2.

4.4 Event Trigger Enhancements

A new feature of the MibADC is that group 1 and the event group have the option to be event triggered, which produces a maximum of two event-triggered groups. A multiplexer allows the event source for both group 1 and the event group to be independently selected from either internal timers or the ADEVT pin.

Programming group 1 for event-triggered mode does not change its priority with respect to other groups when multiple groups are ready to be converted. It simply changes the behavior of group 1 so that group 1 is not ready to be converted immediately after ADISR1 is written; rather, it is ready for conversion only after the event which is programmed to trigger group 1 occurs.

The arbitration among groups (including behavior with freeze enable set) remains the same when group 1 is programmed for event-triggered mode and an event is pending. This behavior is fairly complex, and is described in more detail in Section 5.

4.5 Buffered Mode Control Registers

Table 3 lists the buffered mode control registers.

Table 3. Buffered Mode MibADC Control Registers

Address	Mnemonic	Name	Description
0x00	ADCR1	Control register 1	Contains control bits for initiating and controlling calibration and conversions
0x04	ADCR2	Control register 2	Contains control bits for initiating and controlling modes and events
0x08	ADSR	Status register	Contains flags showing the status of conversion activities
0x0C	ADEISR	Event group input-channel select register	Controls start of conversion of selected channels based on ADEVT event

Table 3. Buffered Mode MibADC Control Registers (Continued)

Address	Mnemonic	Name	Description
0x10	ADISR1	Group 1 input-channel select register	Controls start of conversion of selected group 1 channels
0x14	ADISR2	Group 2 input-channel select register	Controls start of conversion of selected group 2 channels
0x18	ADCALR	Calibration and offset error correction register	Receives calibrated data from ADC core
0x20 - 0x3F	ADBUFE	Event group buffer	FIFO 1 read location, event group
0x90	ADEMBUFE	Event Group Emu Buffer	FIFO 1 read location, event group - *No Side Effect to Read
0x40-0x5F	ADBUF1	Group 1 Buffer	FIFO 2 read location, group 1
0x94	ADEMBUF1	Group 1 Emu Buffer	FIFO 2 read location, group 1 - *No Side Effect to Read
0x60-0x7F	ADBUF2	Group 2 Buffer	FIFO 3 read location, group 1
0x98	ADEMBUF2	Group 2 Emu Buffer	FIFO 3 read location, group 1 - *No Side Effect to Read
0x9C	ADINR	Data Input Register	Contains last converted value on the ADVENT pins
0xA0	ADCPC	Pin Control Register	Controls ADEVT pin
0xB0	ADSAMPEV	Sample Control Register Event	Event group sample time
0xB4	ADSAMP1	Sample Control Register 1	Group 1 sample time
0xB8	ADSAMP2	Sample Control Register 2	Group 2 sample time
0xBC	ADBCR1	Buffer Control Register 1	Buffer enable/disable, allocation
0xC0	ADBCR2	Buffer Control Register 2	Buffer end pointer/size control
0xC4	ADBCR3	Buffer Control Register 3	Buffer DMA and interrupt enable bits
0xC8	ADBST	Buffer Status Register	Buffer empty, overrun, and interrupt flags
0xCC	ADTHREV	Event Group Interrupt Threshold	Sets fill level of event group FIFO before interrupt generated
0xD0	ADTHRG1	Group Interrupt 1 Threshold	Sets fill level of group 1 FIFO before interrupt generated
0xD4	ADTHRG2	Group 2 Interrupt Threshold	Sets fill level of group 2 FIFO before interrupt generated
0xD8	ADEVTSRC	Event Source Register	Selects extended event trigger sources for ADEVT group and group 1, and enables group 1 to act as an event-triggered group

4.5.1 Buffered Mode FIFOs

In buffered mode, the 16 digital result registers and their corresponding 16 emulation registers are replaced with 3 FIFOs and an emulation version of each.

Each buffer is readable from more than one register address (aliased) to allow effective use of the CPU capability to load multiple instructions to empty the buffer. For example, a read to any address between offset 0x20 and 0x3F pulls one conversion result from the event group buffer.

Also, for debug purposes, each buffer has an emulation address that returns the next conversion result from the buffer without removing the result from the buffer itself. For example, reading from offset 0x90 returns the next result in the event group buffer, but does not actually remove that result from the buffer or change the amount of data held in the buffer.

Only the MibADC module itself writes to the FIFOs. Data is written to the specific FIFO for a particular conversion group when a conversion from that group is complete.

Data can be read out of a FIFO either by the CPU or the DMA controller. The MibADC cannot distinguish between the two.

Reading from an empty FIFO results in a data value with bit 15 (EMPTY) set. This bit can be checked to verify that the conversion data is indeed valid.

Each FIFO may be read (either by the CPU or DMA) at the same time a new conversion result is stored in it by the MibADC. This feature is supported in hardware for simplicity and no additional wait states are incurred in this situation.

The MibADC works with the DMA in request mode generating one DMA request for each piece of data moved into a FIFO. Therefore, when a FIFO is serviced by the DMA controller, an empty buffer read should not occur during normal operation. However, the EMPTY flag can be checked as a precaution.

If the CPU is used to read from a FIFO, the load multiple (LDM) instruction enables the loading of multiple registers from memory with back-to-back read operations. The EMPTY flag saves the CPU from polling the DMA controller between each data read to determine if there is more data in the FIFO.

Finally, a FIFO overrun occurs when a FIFO is full and the MibADC attempts to write data into the FIFO while no data is being read from the FIFO. If this occurs, the FIFO blocks the write (does not allow an overrun to occur) and goes into an overrun state. In the overrun state, no new data can be written to the FIFO (new conversions are lost), but the data already in the FIFO can

be read either by the CPU or DMA. An interrupt request is generated, and the CPU must re-initialize the conversion group and its FIFO.

4.5.2 FIFO Channel Identifier

Optionally, as each channel is converted and its digital result is stored in the FIFO, the channel number can be stored along with the digital result in bit positions [13:10]. This is useful for verifying that the data read from a FIFO is the conversion result of the expected channel.

This mode is optional and must be enabled by setting the CHID bit in the ADBUFCR2 register.

4.5.3 FIFO 8-Bit Mode

Some applications can use only the eight most significant bits (MSBs) of the A/D result. To make using the MibADC in this way more efficient, the MibADC can automatically shift the conversion result right by two bits as the result is read out of a FIFO. This shift eliminates the need for the application program to shift each piece of data and results in faster execution.

The FIFO 8-bit mode is optional, and must be enabled by setting the G18BIT, G28BIT, and EV8BIT control bits in the ADBUFCR2 register. It is selectable on a per-FIFO basis. This option is functional only in FIFO mode.

5 Modes of Operation

This section provides a general discussion of the four operating modes of the TMS470 MibADC.

- **Conversion mode**: The normal active mode for converting the selected external input voltage according to the configuration set in ADCR2 and according to the timing selected by PS.2:0 and ACQ.1:0 in ADCR1.4:0 or, alternatively, ADSAMPEV.7:0, ADSAMPG1.7:0, and ADSAMPG2.7:0. See Section 5.1, *Conversion Mode* for more details.
- **Self-test mode**: An active mode that inserts a failure-detection step before the normal conversion flow is entered. Self-test mode is enabled by writing a 1 to the SELF TST bit (ADCR1.9). See section 5.2, *Self-Test Mode*, for more details.
- **Calibration mode**: A special active mode that allows an internal reference voltage to be connected to the ADC core input. While the calibration is selected, the input channel multiplexer is disabled and only the reference is available at the converter input. The calibration mode is enabled by writing a 1 to the CAL EN bit (ADCR1.13). See section 5.3, *Calibration Mode*, for more details.
- **Power-down mode**: Inactive mode in which the ADC internal clock is stopped, leaving the module in a static state. After power-down mode is released, time $t_{d(PU-ADV)}$ (see the device-specific data sheet for specific timing) is required for the MibADC to stabilize. See section 5.4, *Power-Down Mode*, for more details.

5.1 Conversion Mode

The MibADC offers two modes of conversion: continuous conversion mode and single conversion mode. Both conversion modes are selectable for each conversion group.

Note: Continuous Conversion for Multiple Groups

The three groups (G1, G2, and EVT) cannot operate together in continuous mode. When this situation occurs, group 2 is automatically reset to single conversion mode, and the G2 MODE (ADCR2.2) bit is cleared to reflect the actual conversion mode of group 2.

There are two conversion modes:

- Single conversion mode

  In single conversion mode, the MibADC converts all selected channels in a group sequentially one time through and then stops.

  The entire conversion sequence, from the acceptance of a group conversion request to the end of the last conversion, is flagged for each group by the corresponding BUSY bit (ADSR.13:11). After single conversion mode is started, the BUSY bit is read as 1 until the conversion of the last selected channel is complete.

- Continuous conversion mode

  In continuous conversion mode, the MibADC module performs a sequential circular conversion of all selected channels in a group from the lowest to the highest selected channel. After the highest channel is converted, the MibADC module checks for groups with pending group conversion requests; the group with the highest priority is converted next unless there are any other conversion requests. In this case, the current group is not converted next, and the continuous group acts like a constant request. Therefore, when two channels are configured as continuous, they alternate conversion; however, new group conversion requests have higher priority than the continuous conversion groups. If there are no higher pending requests, continuous conversion mode starts over with the lowest channel.

  The entire conversion sequence, from the acceptance of a group conversion request to the end of the last conversion, is flagged for each group by the corresponding BUSY bit (ADSR.13:11). After continuous conversion mode is started, the BUSY bit reads 1 as long as continuous conversion mode is active.

  Continuous conversion mode can be stopped in one of two ways. To stop conversions immediately, clear the input channel-select register (ADEISR, ADISR1, or ADISR2) to 0x0. To stop the continuous conversions after the current conversion is complete, clear the mode bit (EV MODE, G1 MODE, or G2 MODE).

5.1.1 Changing an Input Channel Select Register During a Conversion

It takes one ICLK period ($t_{c(ICLK)}$) for the MibADC to skip an unselected channel. However, the MibADC monitors for the next channel to convert while it is converting the current channel; the ICLK cycles are carried out in the background. Typically, the ICLK periods that are required to skip an unselected channel are not visible after the first selected channel.

When you change the group definition in the input channel-select register during conversion, the behavior of the MibADC module is determined by the group that is changed during the conversion.

The EVT group is configured by writing to the event group input channel select register; the EVT group is initiated by an edge with polarity that matches the set polarity on the ADEVT pin. Thus, when you change the group definition in the input channel select register during conversion, the MibADC finishes converting the current channel and then loads the new group into the input channel select register. The MibADC then begins monitoring for the next input channel to convert from the next channel to have been converted in the original group.

5.1.2 Changing Input Channel Selections During an Event Conversion

Suppose the original EVT group enables channels 0, 1, 8, and 13, and you change the group definition to 1, 7, and 10 while channel 1 is being converted. While the MibADC is converting channel 1, it monitors for the next selected channel, which is channel 8. As soon as the conversion of channel 1 is finished, the MibADC loads the new group into the input channel select register. The MibADC has already checked channels 2 through 7 before loading the new group, so now the MibADC begins checking for channels to convert beginning at channel 8. The next channel to be converted, therefore, is channel 10, and this conversion cycle misses channel 7 completely.

Group 1 and group 2 are configured **and** initiated by writing to the group input channel select register. Thus, when you change the group definition in the input channel select register during conversion, the MibADC finishes converting the current channel and then loads the new group into the input channel select register. The MibADC restarts the group conversion (including priority resolution) and then begins monitoring for the next input channel to convert.

5.1.3 Changing Input Channel Selections During a Group 1 or 2 Conversion

Suppose the original group (G1 or G2) enables channels 0, 1, 8, and 13, and you change the group definition to 1, 7, and 10 while channel 1 is being converted. While the MibADC is converting channel 1, it checks to find the next selected channel, which is channel 8. After channel 1 is converted, the MibADC loads the new group into the input channel select register as a new group conversion request. The MibADC checks channel 0 and finds that it is not selected. It checks channel 1 and converts channel 1 followed by channels 7 and 10.

Note: EVT Trigger During an Ongoing EVT Conversion Restarts EVT Conversion

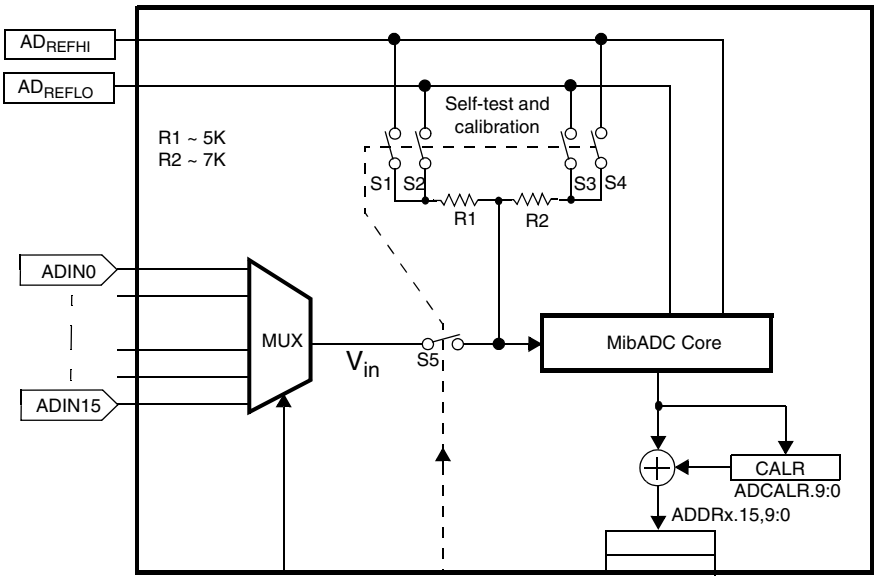
Although changing the configuration of the EVT group does not restart the conversion, a second valid edge triggers a new conversion. The MibADC restarts the group conversion (including priority resolution) then begins monitoring for the next input channel to convert.

5.2 Self-Test Mode

Self-test mode is enabled by setting the SELF TST bit (ADCR1.9). Any conversion type (continuous or single conversion, freeze enabled or non-freeze enabled, interrupts enabled or disabled) can be performed in this mode.

In normal mode, setting the self-test mode while a conversion sequence is in process can corrupt the current channel conversion results. However, the next channel in the sequence is converted correctly during the additional self-test cycle. The logic associated with both self-test and calibration is shown in Figure 6.

Figure 6. Self-Test and Calibration Logic



In self-test mode, a test voltage defined by the HILO bit (ADCR1.10) is provided to the MibADC core input through a resistor (see Table 4). To change the test source, this bit can be toggled before any single conversion mode

request. Changing this bit while a conversion is in progress *can* corrupt the results if the source switches during the acquisition period.

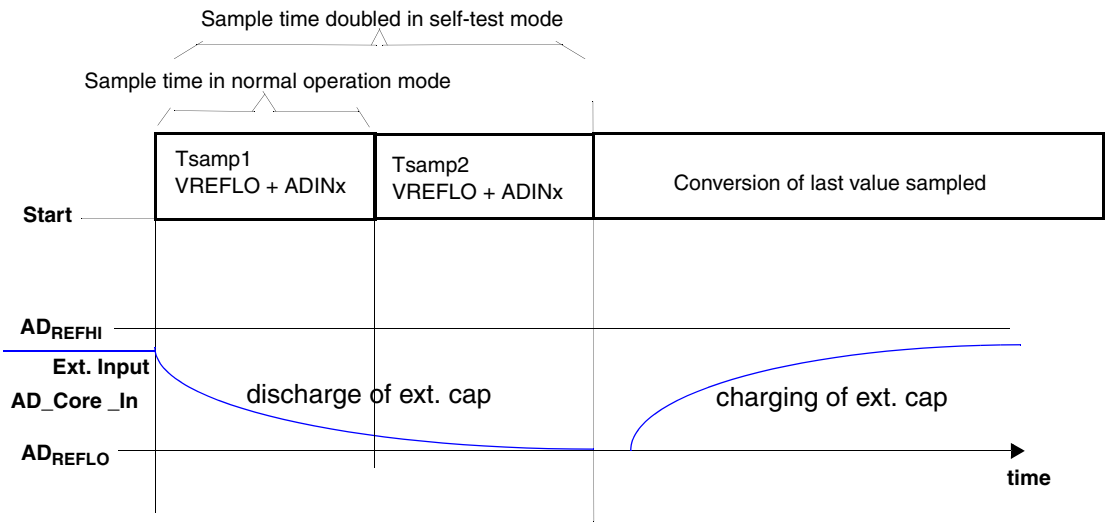
Table 4. Self-Test Reference Voltages(1)

SELF TST	HILO	S1	S2	S3	S4	S5	Reference Voltage
1	0	0	1	1	0	1	AD_{REFLO} via R1 || R2 connected to V_{in}
1	1	1	0	0	1	1	AD_{REFHI} via R1 || R2 connected to V_{in}
0	X	0	0	0	0	1	V_{in}

(1) Switches refer to Figure 6.

Conversions in self-test mode are invoked by writing to ADISR1 or ADISR2, or upon an external event. The conversion starts according to the configuration set in the ADCR2 register. The acquisition time for each conversion is extended to twice the normal configured acquisition time. The selected reference voltage (see Table 4) and the input voltage from ADINx are both connected to the ADC internal sampling capacitor throughout this extended acquisition time. Figure 7 shows the self-test mode timing when the AD_{REFLO} is chosen as the reference voltage for the self-test mode conversion. It also assumes an external capacitor connected to the ADC input channel.

Figure 7. Timing for Self-Test Mode



The digital results are stored in the registers associated with the selected channels.

5.2.1 Use of Self-Test Mode to Determine Open/Short on ADC Input Channels

The self-test mode is primarily designed for detecting opens and shorts on the ADC input channel. This section describes the procedure for making this determination.

1) Perform a single conversion in normal mode for the ADC input channel under test. Call the conversion result Vn.
2) Perform a single conversion in self-test mode with AD_{REFHI} as the reference voltage. Call this conversion result Vu.
3) Perform a single conversion in self-test mode with AD_{REFLO} as the reference voltage. Call this conversion result Vd.

The results can be interpreted using the following table.

Table 5. Determination of ADC Input Channel Condition

Normal Conversion Result, Vn	Self-test Conversion Result, Vu	Self-test Conversion Result, Vd	Pin Condition
Vn	Vn < Vu < AD_{REFHI}	AD_{REFLO} < Vd < Vn	Good
AD_{REFHI}	AD_{REFHI}	approx. AD_{REFHI}	Shorted to AD_{REFHI}
AD_{REFLO}	approx. AD_{REFLO}	AD_{REFLO}	Shorted to AD_{REFLO}
Unknown	AD_{REFHI}	AD_{REFLO}	Open

5.3 Calibration Mode

The application program can activate a calibration sequence any time self-test mode is disabled (SELF TST = 0). This calibration sequence includes the conversion of an embedded calibration reference voltage followed by the calculation of an offset error correction value.

Note: Disable Self-Test Mode Before Calibration

To avoid errors during the calibration operation, self-test mode must *not* be enabled during a calibration sequence. In addition, to ensure accurate results, calibrate the ADC in an environment with minimum noise.

Calibration mode is enabled by setting the CAL EN bit (ADCALCR.0). In this mode, the current ADC activity is frozen, depending on the freeze option of each group. The input multiplexer is disabled and only the reference voltage is connected to the ADC core input. The multiplexer is disconnected (S5 of

Figure 6 is open). In addition, the digital result issued from a conversion is output from the ADC core to the calibration and offset error correction register, ADCALR. The digital result register frame is not affected and retains the previous data.

When calibration mode is disabled, the ADC restarts operations from where it was frozen.

Note: Disable all groups' conversions before enabling calibration

The calibration mode must be enabled only when no other group conversion is active.

5.3.1 Calibration Conversion

The BRIDGE EN and HILO bits (ADCALCR.9:8) control the voltage to the calibration reference device shown in Figure 6. The positions of the switches in calibration mode are listed in Table 6.

Table 6. Calibration Reference Voltages†

CAL EN	BRIDGE EN	HILO	S1	S2	S3	S4	S5	Reference Voltage
1	0	0	1	0	1	0	0	$(AD_{REFHI}*R1 + AD_{REFLO}*R2) / (R1 + R2)$
1	0	1	0	1	0	1	0	$(AD_{REFLO}*R1 + AD_{REFHI}*R2) / (R1 + R2)$
1	1	0	0	1	1	0	0	AD_{REFLO}
1	1	1	1	0	0	1	0	AD_{REFHI}
0	X	X	0	0	0	0	1	V_{in}

† The state of the switches in this table assumes that self-test mode is not enabled.

When CAL ST (ADCALCR.16) is set, a calibration conversion is started. The voltage source selected via the bits BRIDGE EN and HILO is converted once (single conversion mode) and the digital result is returned to the calibration and correction register, ADCALR, where it can be read by the CPU. The CAL ST bit acts as a flag and must be polled by the CPU. It is held set during the conversion process and automatically clears to indicate the end of the reference voltage conversion.

Note: There is no interrupt to signal the end of the calibration conversion. The CPU must poll the CAL_STR bit to determine the end of the calibration conversion.

After the CAL ST bit is set by the application program, it can only be reset by the end of the ongoing conversion generated by the ADC core. If the calibration conversion is interrupted (CAL EN bit is cleared), the CAL ST bit is held at 1 until a new calibration conversion has been set and completed. Setting the CAL ST bit while calibration is disabled (CAL EN = 0) has no effect; however, in this situation, setting CAL EN immediately starts a calibration conversion. When the calibration conversion is interrupted by an ADC enable (ADC EN = 0, CAL EN = 1, and CAL ST = 1), a new conversion is automatically restarted as soon as the ADC enable bit is released (ADC EN = 1).

5.3.2 Calibration and Offset Error Correction Sequences

The number of measurements and the source to measure for an ADC calibration are application dependent. The CAL ST bit must be set for each calibration source to be measured. While calibration mode is enabled, any available calibration sources can be converted according to the BRIDGE EN and HILO bits (see Table 6). The digital results of the calibration measurements should be read from ADCALR by the application after each reference conversion so that a correction value can be computed and written back into ADCALR.

When the application has the necessary calibration data, it should compute the offset error correction value and load it into the calibration and correction register, ADCALR. After the CAL EN bit is cleared, normal conversion mode restarts, continuing from where it was frozen, but with the addition of self-correction data.

In normal mode, the self-correction system adds the correction value stored in ADCALR to each digital result before it is written to the respective group's FIFO.

The basic calibration routine is as follows:

1) Enable calibration via CAL EN (ADCALCR.0).
2) Select the voltage source via BRIDGE EN and HILO (ADCALCR.9:8).
3) Start the conversion with CAL ST (ADCALCR.16).
4) Wait for CAL ST to go to 0.
5) Get the results from ADCALR and save to memory.
6) Loop to step 2 until the calibration data is collected, then continue.
7) Compute the error correction value using calibration data saved in memory.
8) Load the ADCALR register with the 2s complement of the computed value.

9) Disable calibration mode.

At this point, the ADC resumes normal operation and corrects each digital result with the value computed and loaded into ADCALR.

Note: Prevent ADC Calibration Data From Being Overwritten

In calibration mode, the conversion result is written to ADCALR which overwrites any previous calibration data; therefore, the ADCALR register must be read before a new conversion is started.

For no correction, a value of 0x0000 must be written to ADCALR. In non-calibration mode, the ADCALR register can be read and written. Any value written to ADCALR in normal mode (CAL EN = 0) is added to each digital result from the ADC core.

5.3.3 Mid-Point Calibration

Because of its connections to the ADC's reference voltage (VrefHi, VrefLo), the precision of the calibration reference is voltage independent. On the other hand, the accuracy of the switched bridge resistor (R1 & R2) relies on the manufacturing process deviation. Consequently, the mid-point voltage's accuracy can be affected due to the imperfections in the two resistors (expected mismatch error is around 1.5%).

The switched reference voltage device has been specially designed to support a differential measurement of its mid-point voltage. This ensuring the accuracy of the mid-point reference, hence the efficiency of the calibration.

The differential mid-point calibration is software controlled; the algorithm (voltage source measurements and associated calculation) is inserted within the calibration software module included in the application program.

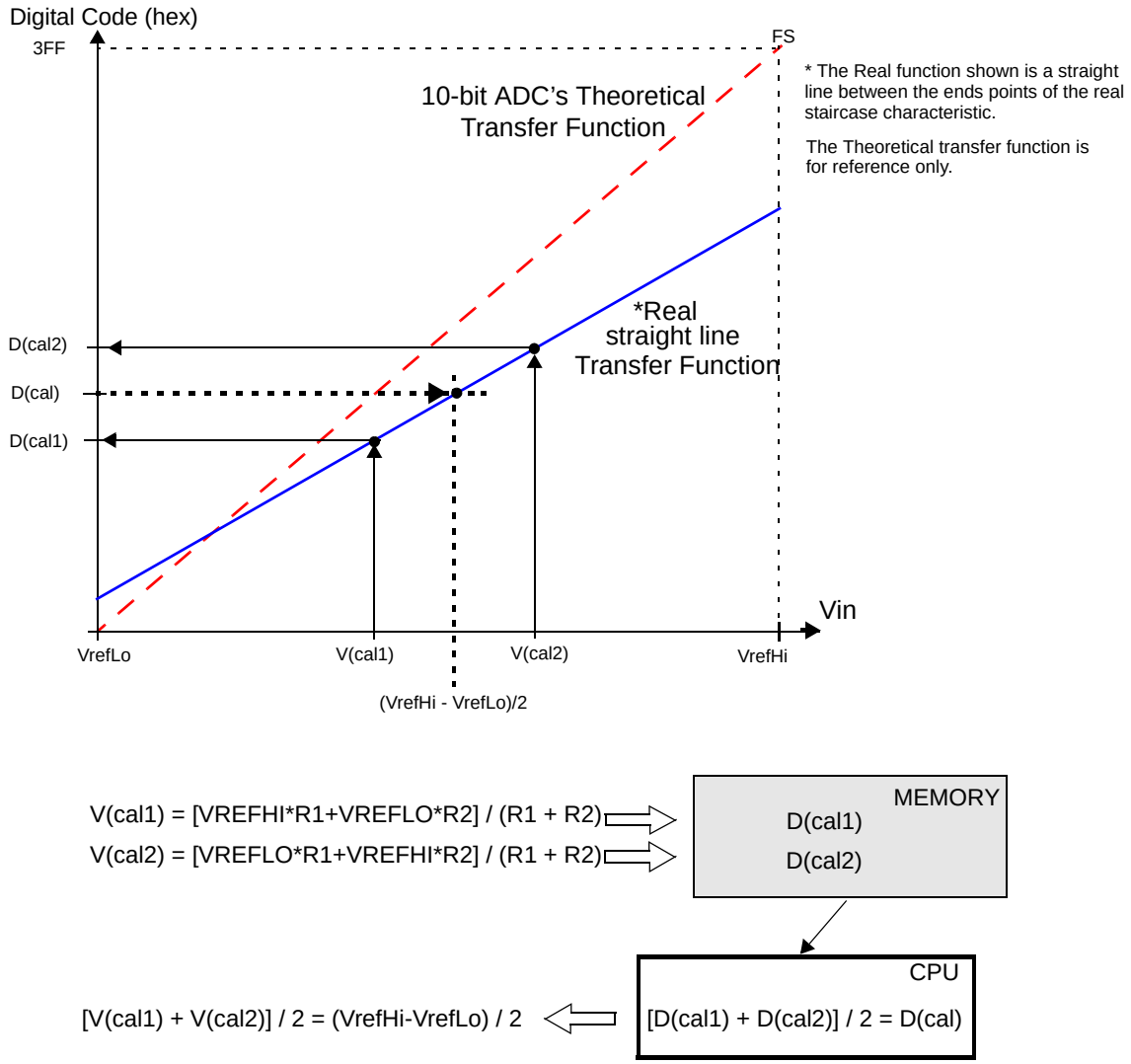
The basic differential mid-point calibration flow is illustrated here after

Step 1: The application program connects the voltage VrefHi to R1 and VrefLo to R2, (BRIDGE_EN=0, HILO=0), launches a conversion of the input voltage V(cal1), and stores the digital result D(cal1) into the memory.

Step 2: Then the application program switches the voltage VrefHi to R2 and VrefLo to R1 (BRIDGE_EN=0, HILO=1), converts this new input voltage V(cal2) and again stores the issued digital result D(cal2) into the memory.

Step 3: The actual value of the real middle point is obtained by computing the average of this two results. [D(cal1)+D(cal2)] /2; The Figure below summarizes the mid-point calibration flow.

Figure 8. Mid-point value calculation



5.4 Power-Down Mode

In power-down mode, the MibADC internal clock is stopped and the module is in a static state. The continuous bias current required for the analog stages is removed and the resistor ladder connected between AD_{REFHI} and AD_{REFLO} is opened. This results in the lowest possible MibADC power consumption. The MibADC enters power-down mode when the PWR DN bit (ADCR1.8) is set.

The MibADC registers continue to be clocked. Setting the PPWNOVR bit (CLKCNTL.7 in the system module) disables the clocks to the control registers of the MibADC module when the power-down bit (PWR DN) of the module is set.

Power-down mode is released by clearing the PPWNOVR bit and then clearing the PWR DN bit. After power-down mode is released, time $t_{d(PU-ADV)}$ is required for the MibADC to stabilize. See the device-specific data sheet for this value.

6 Conversion Priority and the Freeze-Enable Bit

This section discusses various priorities and the freeze-enable bit.

6.1 Group Priority

A group conversion request requires the group to be configured and triggered. For group 1 (G1) and group 2 (G2), the groups are configured and triggered by writing to the group select register: ADISR1 or ADISR2. For the event group, writing to the event group select register, ADEISR, configures the group. After the group is configured, a valid edge on the ADEVT pin triggers the conversion.

The group priority determines the order of conversion in cases of multiple group conversion requests; when one conversion is requested, priority is irrelevant. There are two primary conditions when priority is considered:

- ❏ Multiple group conversion requests occur during an ongoing conversion.
- ❏ A group conversion request coincides with an external event trigger.

The priority of groups is (from highest priority to lowest priority):

1) Event (EVT)
2) Group 1 (G1)
3) Group 2 (G2) (See Section 6.1.1, *Simple Priority*.)

In continuous conversion mode, the conversion takes place as specified by the priority. Continuous conversion mode acts like a continually pending group conversion request with the exception that the MibADC can distinguish explicit group conversion requests (the explicit group conversion request is triggered by writing to the G1 or G2 group select register or by a valid edge on the ADEVT pin) from the continuous request. The MibADC module converts explicit group requests with higher priority than a pending continuous conversion. (See Section 6.1.2, *Explicit Conversion Requests and Continuous Conversion Requests*.) For two continuous conversions, the groups alternate (See Section 6.1.3, *Alternating Conversion of Continuous Groups*.)

The freeze bit affects the outcome of priority contentions, but the freeze option is separate from priority. When the freeze bit is set for a group, the frozen group allows another group conversion request to interrupt the active (frozen) group. (See Section 6.1.4, *Example Priority With Freeze*.)

6.1.1 Simple Priority

If an event group conversion is active (single conversion and FRZ EV = 0) and group 2 and group 1 conversions are requested, then when the event group is finished, group 1 is converted. Then group 2 is converted.

6.1.2 Explicit Conversion Requests and Continuous Conversion Requests

If group 1 is active (continuous conversion and FRZ G1 = 0) and group 2 conversion is requested, when the current conversion of group 1 is finished, group 2 is converted. The explicit conversion request of group 2 has higher priority than the pending continuous conversion request of group 1.

6.1.3 Alternating Conversion of Continuous Groups

If group 1 is being converted (single conversion and FRZ G1 = 0) and group 2 (continuous and FRZ G2 = 0) and the event group (continuous and FRZ EV = 0) request conversion, the event group is converted, and group 2 is converted; the conversions continue to alternate in this order.

6.1.4 Example Priority With Freeze

If two group conversion requests are active, then the priority dictates which group is converted first. However, if the group that wins the priority contention is freeze enabled, it begins its conversion and is interrupted by the second group before any channels are converted. Thus, the group with lower priority is converted before the higher priority, freeze-enabled group.

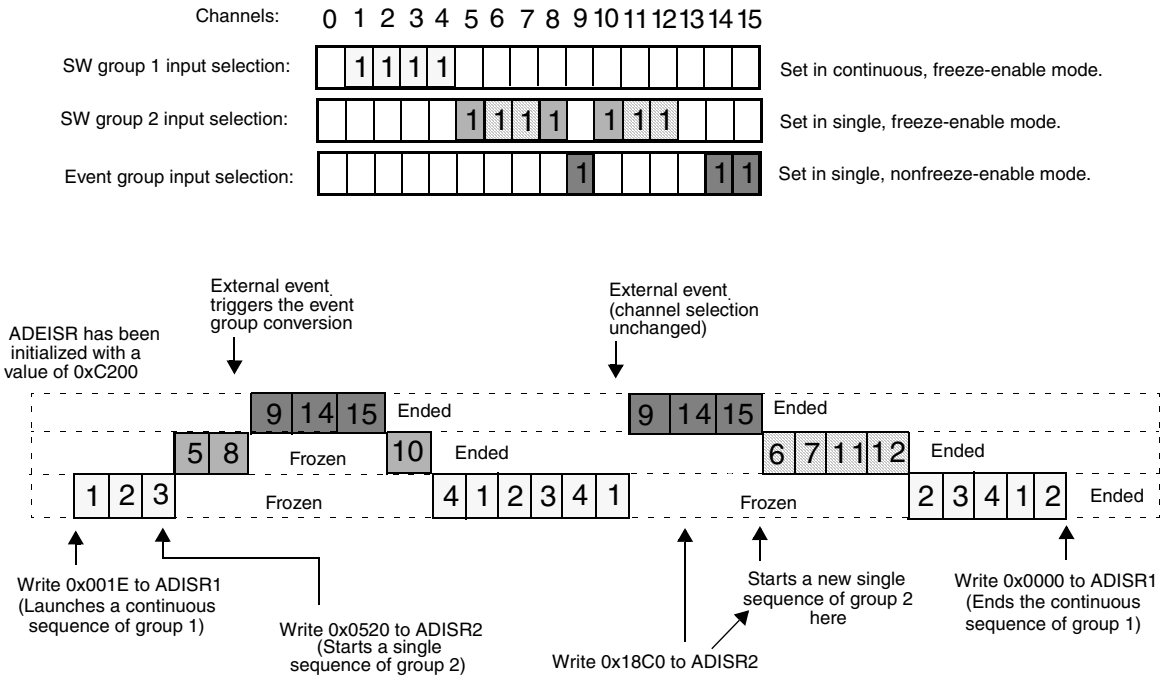
The freeze-enable bit for each conversion group, FRZ EV (ADCR2.10), FRZ GR1 (ADCR2.4), and FRZ GR2 (ADCR2.1), allows conversions to be prioritized according to the following rules:

- When the freeze-enable bit for a group is set, the priority of that group is *lowered,* which freezes a conversion after completion of the ongoing channel *if* another higher priority group requests conversion.
- When all conversions for the new higher-priority group are complete, the frozen group is again enabled, and the conversion resumes from the point where it was frozen. For an example, see Section 6.2, *Illustration: Multigroup Conversion Sequence Using Freeze Bits*, on page 38. During the frozen time of the conversion, the frozen condition of the selected group is flagged by the conversion stop bit (STOP) in register ADSR.

6.2 Illustration: Multigroup Conversion Sequence Using Freeze Bits

Figure 9 shows all three groups in a sequence together. The freeze bits have been set to allow the newer group request to freeze the group currently in conversion.

Figure 9. Example of Conversion Sequence Using the Three Groups



Group 1 is programmed to convert channels 1 through 4 in continuous (repeating) mode, and is freeze enabled, which means it has a lower conversion priority than either group 2 or the event group.

During the conversion of channel 3, group 2 is programmed to convert channels 5, 8, and 10 in single-conversion (one-pass) mode, and is also freeze enabled. Because the freeze bit of group 1 is *also* set, group 2 implicitly has higher priority than group 1. However, the priority of group 2 is lower than the priority of the event group. As soon as the ADISR register of group 2 is written, the next conversion slot is given to group 2 (channel 5), and group 1 is frozen (between channels 3 and 4).

The event group conversion request occurs sometime during the conversion of channel 8 (from group 2). Immediately following the conversion of channel 8, conversion of channel 9 from the event group starts.

When conversion of all channels of the event group is complete (9, 14, and 15), group 2 is unfrozen, and the conversion of the last channel of group 2 is completed.

Now, group 1 goes back to continuous conversion of channels 1 through 4 and resumes where it left off with channel 4. This conversion continues until a second event group occurs where channels 9, 14, and 15 are once again converted.

Sometime during the conversion of the higher priority event group (channels 9, 14, or 15) user software writes to the ADISR2 register requesting that channels 6, 7, 11, and 12 be converted. Because group 2 has priority over group 1 but not over the event group, conversion of channel 6 does not start until the conversion of channel 15 is finished.

When all of the new group 2 channels have been converted, the conversion of group 1 resumes.

Group 1 is terminated when ADISR1 is cleared (no channels selected) by writing 0x0000 to it.

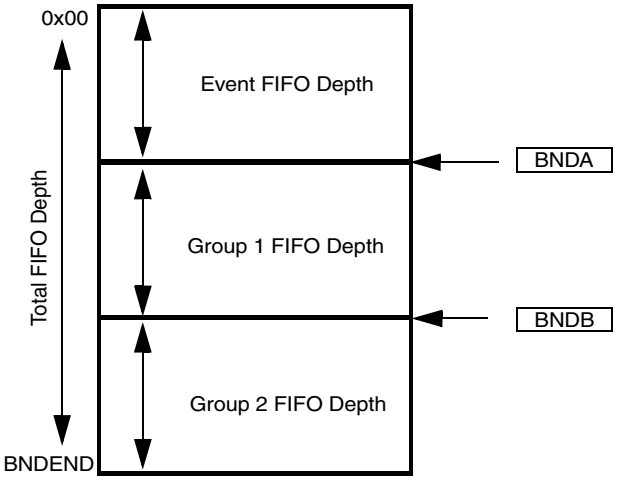
7 MibADC FIFO Configuration

The buffered MibADC includes a fixed amount of FIFO memory that can be divided by the application into an event group FIFO, a group 1 FIFO, and a group 2 FIFO.

Because the number of conversion channels assigned to each conversion group and the rate at which each group is converted varies from application to application, the depth of each of the FIFOs is programmable and is controlled by the following two registers: ADBCR1 and ADBCR2.

- ADBCR1 controls two pointers, BND1 and BNDB. They are used to partition the total FIFO memory available into three sections as shown in Figure 10.

Figure 10. FIFO Implementation



- ADBCR2 contains a 3-bit field that represents the total FIFO depth:

BNDEND[2:0] = 000 End Boundary = 16 Words
BNDEND[2:0] = 001 End Boundary = 32 Words
BNDEND[2:0] = 010 End Boundary = 64 Words
BNDEND[2:0] = 011 End Boundary = 128 Words
BNDEND[2:0] = 100 End Boundary = 192 Words
BNDEND[2:0] = 101 End Boundary = 256 Words
BNDEND[2:0] = 110 End Boundary = 384 Words
BNDEND[2:0] = 111 End Boundary = 512 Words

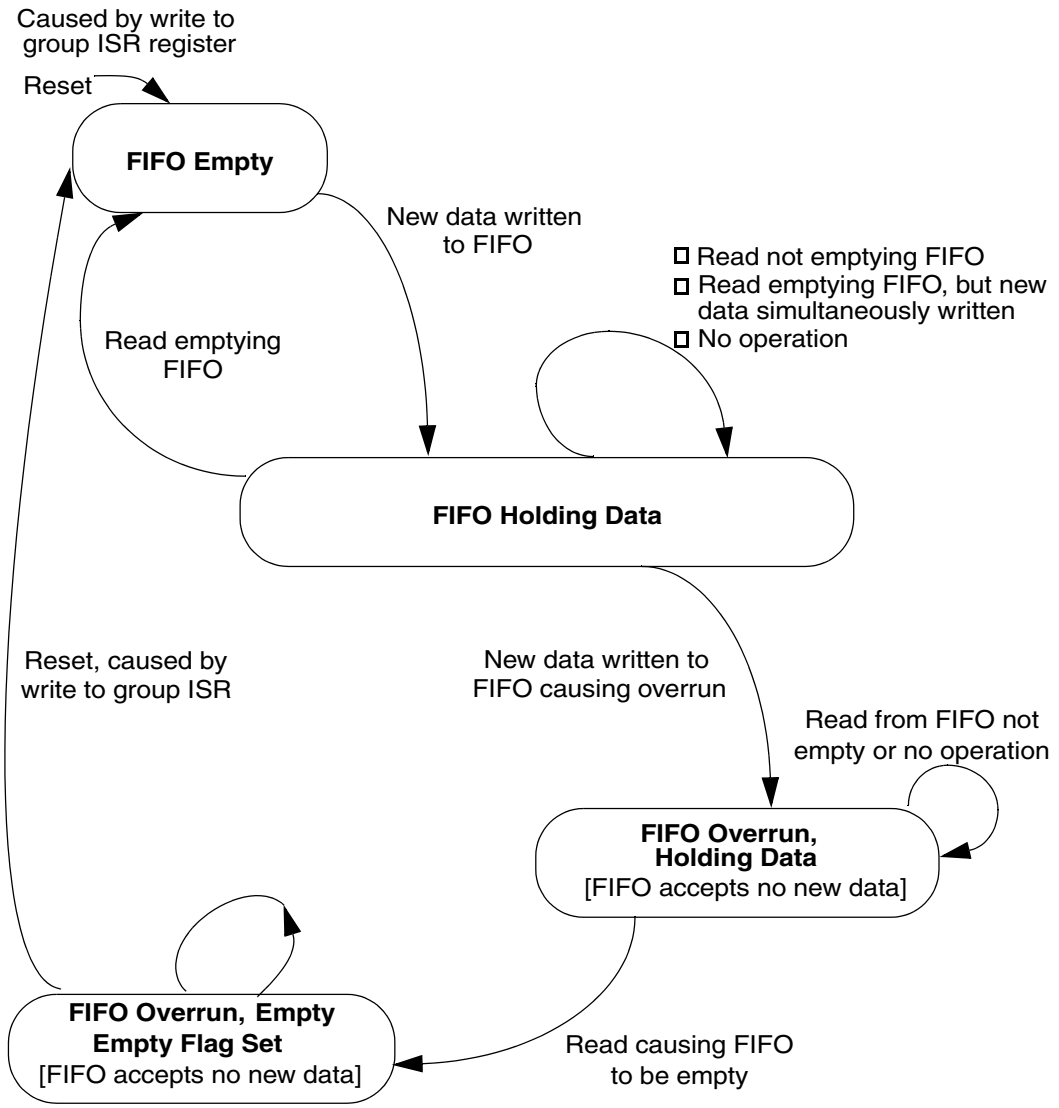
7.1 FIFO State Machine

Each FIFO includes a state machine that manages the contents of the FIFO and flags overruns and empty FIFO conditions, as shown in Figure 11.

The four states of each FIFO buffer are described as follows:

- **FIFO Empty**: There is no data in the FIFO, and it is ready to accept conversion results from the MibADC.
- **FIFO Holding Data**: In this state, data has been stored in the FIFO by the MibADC core, but some or all of the data has not yet been read out of the FIFO by either the processor or the DMA controller.
- **FIFO Overrun, Holding Data**: This state is caused by the MibADC attempting to write to the FIFO new data that requires more storage than has been allocated to the FIFO in the ADBCR1 and ADBCR2 registers. The FIFO discards all new conversion results. All results stored before the overrun occurred are maintained and can be read out before the FIFO is reset. (This is optional; the FIFO can also be reset and these results discarded.)
- **FIFO Overrun, Empty**: All data has been read out after an overrun has occurred, and the FIFO must be reset.

Figure 11. FIFO State Machine: State Diagram



8 Interrupt Generation

8.1 Interrupt Generation in Compatibility Mode

This section describes interrupt generation in compatibility mode only. For a description of interrupt generation in buffered mode, refer to Section 8.2, *Interrupt Generation in Buffered Mode*, on page 44.

The MibADC has three interrupt sources, one for each group. Each has an associated interrupt request line to the CPU:

- Setting the ENA G1 INT bit (ADCR2.3) enables interrupts at the end of a conversion sequence that was initiated by a write to ADISR1. The G1 END bit (ADSR.2), which is set after all selected channels have been converted, is used to flag the interrupt request if interrupts are enabled.
- Setting the ENA G2 INT bit (ADCR2.0) enables interrupts at the end of a conversion sequence that was initiated by a write to ADISR2. The G2 END bit (ADSR.1), which is set after all selected channels have been converted, is used to flag the interrupt request if interrupts are enabled.
- Setting the ENA EV INT bit (ADCR2.9) enables interrupts at the end of a conversion sequence initiated by an external event at the ADEVT pin or by software through the ADPCR register. The EV END bit (ADSR.0), which is set after all selected channels have been converted, is used to flag the interrupt request if interrupts are enabled.

In all three interrupt request cases, the interrupt request line is maintained to the active level as long as interrupts remain enabled and the matching END bits stay set. The CPU must acknowledge the interrupt and clear the flag, which releases the request line to a non-active level. See section 11.3, *AD Status Register (ADSR)*.

Note: Clear End Flags Before Enabling Interrupts

If the enable interrupt bit is set when the END flag is already set, then an interrupt to the CPU is requested.

No hardware priority is fixed within these three sources. MibADC interrupt arbitration is managed by the central interrupt manager (CIM) in the system module. In the 470R1x device, the interrupt priority relative to other peripherals is fixed. See the specific device data sheet for details on this priority. The main priority level (FIQ or IRQ) for each source is programmable in the CIM (REQMASK and FIRQPR in the system module).

8.2 Interrupt Generation in Buffered Mode

A threshold register exists for each FIFO. The threshold is initialized to the data count at which the CPU should be interrupted.

Each time data is stored in the FIFO, the threshold counter decrements toward 0. Each time data is removed from the FIFO, the threshold counter is incremented toward its initial count. Simultaneous read and write operations from the FIFO leave the threshold counter unchanged.

The threshold counter can decrement past 0 and become negative. It should always increment back to its original value when the FIFO is emptied. To determine how many samples are in the FIFO at a given moment, the threshold counter can be subtracted from the original threshold count.

Whenever the threshold counter transitions from +1 to 0, it sets the buffer interrupt flag, and the CPU is interrupted if the buffer interrupt enable bit is set. The CPU clears the interrupt flag after emptying the FIFO.

The interrupt flag is not set when the threshold counter stays at 0 or transitions from -1 to 0.

The buffered mode interrupt request works in parallel with the existing group-based interrupt generation.

8.2.1 Interrupt Threshold Registers

One interrupt threshold register serves each group FIFO. The register determines how much data must be in the FIFO before the CPU is interrupted. This feature significantly reduces CPU load when interrupts are used to read the FIFO.

9 DMA Request Generation

DMA requests are generated for *request mode* transfers. The first DMA request is generated after the FIFO leaves the EMPTY state and contains data. Subsequently, when the FIFO contains data, a new DMA request is generated after the DMA reads a data word out of the FIFO, if this read does not cause the FIFO to be empty. The DMA request occurs on the very next cycle after the FIFO read.

10 Control Registers Summary

The MibADC has control registers in both compatibility mode and buffered mode.

- ❏ Figure 12 shows registers that work in compatibility mode only.
- ❏ Figure 13 shows registers that work in buffered mode only.
- ❏ Figure 14 shows registers that work in both compatibility mode and buffered mode.

All registers in the MibADC module have 16-bit access and are 32-bit, word-aligned; *only* 16-bit and 32-bit accesses are allowed. Specific bit descriptions are discussed in the following subsections.

The MibADC module contains six registers for controlling MibADC operations. All registers can be read by the CPU at any time without affecting an ongoing conversion or the MibADC accuracy. Writing to the control registers (ADCR1 and ADCR2) or to the input-select registers (ADEISR, ADISR1, and ADISR2) while the corresponding group is being converted can disturb the ongoing channel conversion; however, the module recovers on the next conversion.

Figure 12. MibADC Control Registers in Compatibility Mode

Offset Address † Register		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	ADCR1	COS	Rsrvd	CAL EN	CAL ST	BRIDGE EN	HILO	SELF TEST	PWR DN	Reserved		ADC EN	ACQ. 1	ACQ. 0	PS.2	PS.1	PS.0
0x04	ADCR2	Reserved				EV MODE	FRZ EV	ENA EV INT	EV EDG SEL	Reserved		G1 MODE	FRZ G1	ENA G1 INT	G2 MODE	FRZ G2	ENA G2 INT
0x08	ADSR	Reserved		G1 BUSY	G2 BUSY	EV BUSY	G1 STOP	G2 STOP	EV STOP	Reserved					G1 END	G2 END	EV END
0x0C	ADEISR	EV.15:0															
0x10	ADISR1	G1.15:0															
0x14	ADISR2	G2.15:0															

Figure 12. MibADC Control Registers in Compatibility Mode (Continued)

0x18	ADCALR	Reserved								CALR.9:0					
0x1C	ADDR0	DTx ST	Reserved							DT0.9:0					
0x20	ADEM DR0	EDTx ST	Reserved							EDT0.9:0					
		28 more 16-bit registers alternating between ADDRxx and ADEMDRxx spaced every 4 bytes													
0x94	ADDR15	DT15 ST R-X	Reserved							DT15.9:0					
0x98	ADEM DR15	EDT15ST R-X	Reserved							EDT15.9:0					
0x9C	ADINR	IN.15:0													
0xA0	ADPCR	Reserved											EVT IN	EVT OUT	Rsrvd	EVT DIR

.

Figure 13. MibADC Control Registers in Buffered Mode

Offset Address †	Register	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	ADCR1	COS	Rsrvd	CAL EN	CAL ST	BRI-DGE EN	HILO	SELF TEST	PWR DN	Reserved		ADC EN	ACQ. 1	ACQ. 0	PS.2	PS.1	PS.0
0x04	ADCR2	Reserved				EV MOD E	FRZ EV	ENA EV INT	EV EDG SEL	Reserved		G1 MOD E	FRZ G1	ENA G1 INT	G2 MOD E	FRZ G2	ENA G2 INT
0x08	ADSR	Reserved		G1 BUSY	G2 BUSY	EV BUSY	G1 STOP	G2 STOP	EV STOP	Reserved					G1 END	G2 END	EV END
0x0C	AD EISR	EV.15:0															
0x10	ADISR1	G1.15:0															
0x14	ADISR2	G2.15:0															
0x18	AD CALR	Reserved					CALR.9:0										
0x20 - 0x3F aliased	AD BUFE	EV EMPT Y	0	EVCHID.3:0			EVDR.9:0										
0x40 - 0x5F aliased	AD BUF1	G1 EMPT Y	0	G1CHID.3:0			G1DR.9:0										
0x60 - 0x7F aliased	AD BUF2	G2 EMPT Y	0	G2CHID.3:0			G2DR.9:0										
0x90	ADEMU BUFE	EV EMPT Y	0	EVCHID.3:0			EVDR.9:0										
0x94	ADEMU BUF1	G1 EMPT Y	0	G1CHID.3:0			G1DR.9:0										
0x98	ADEMU BUF2	G2 EMPT Y	0	G2CHID.3:0			G2DR.9:0										

Figure 13. MibADC Control Registers in Buffered Mode (Continued)

0x9C	ADINR	IN.15:0														
0xA0	ADPCR	Reserved											EVT IN	EVT OUT	Rsrvd	EVT DIR
0xB0	AD SAMP EV	SEN	Reserved							EVACQ.7:0						
0xB4	AD SAMP1	Reserved								G1ACQ.7:0						
0xB8	AD SAMP2	Reserved								G2ACQ.7:0						
0xBC	ADB CR1	BUFE N	BNDA.6:0							BNDB.7:0						
0xC0	ADB CR2	Reserved													BNDEND.2:0	
0xC4	ADB CR3	EV DMA EN	G1 DMA EN	G2 DMA EN	EV OVR INT EN	G1 OVR INT EN	G2 OVR INT EN	EV BUF INT EN	G1 BUF INT EN	G2 BUF INT EN	Reserved				EV 8BIT	G1 8BIT	G2 8BIT	CHID
0xC8	AD BUFST	Reserved					EV INT FLAG	G1 INT FLAG	G2 INT FLAG	RES	EV OVR	G1 OVR	G2 OVR	Res	EV EMPT Y	G1 EMPT Y	G2 EMPT Y	
0xCC	AD THREV	Sign Extension								EVTHR.9:0						
0xD0	G1 THREV	Sign Extension								G1THR.9:0						
0xD4	G2 THREV	Sign Extension								G2THR.9:0						
0xD8	AD EVTSR C	Reserved							G1 ENA	G1 EDG SEL	Res	G1SRC.1:0		Res		EVSRC.1:0

Figure 14. MibADC Control Registers Available in Both Compatibility and Buffered Modes

Offset Address[1] Register		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	AD CR1	COS	Rsrvd	CAL EN	CAL ST	BRI-DGE EN	HILO	SELF TEST	PWR DN	Reserved		ADC EN	ACQ. 1	ACQ. 0	PS.2	PS.1	PS.0
0x04	AD CR2	Reserved				EV MOD E	FRZ EV	ENA EV INT	EV EDG SEL	Reserved		G1 MOD E	FRZ G1	ENA G1 INT	G2 MOD E	FRZ G2	ENA G2 INT
0x08	ADSR	Reserved		G1 BUSY	G2 BUSY	EV BUSY	G1 STOP	G2 STOP	EV STOP	Reserved					G1 END	G2 END	EV END
0x0C	ADE-ISR	EV.15:0															
0x10	AD ISR1	G1.15:0															
0x14	AD ISR2	G2.15:0															
0x18	AD CALR	Reserved						CALR.9:0									
0x9C	ADIN R	IN.15:0															
0xA0	ADPC R	Reserved												EVT IN	EVT OUT	Rsrvd	EVT DIR

11 Detailed Description of Control Registers

The MibADC registers are described in the following sections.

11.1 AD Control Register 1 (ADCR1)

This register is available in compatibility and buffered modes. Figure 15 and Table 7 describe this register.

Figure 15. AD Control Register 1 (ADCR1)

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	COS	Rsrvd	CAL EN	CAL ST	BRIDGE EN	HILO	SELF TST	PWR DN	Reserved		ADC EN	ACQ.1:0		PS.2:0		
	RW-0	U	RW-0	RS-0	RW-0	RW-0	RW-0	RW-0	U		RW-0	RW-0		RW-0		
	R = Read; W = Write; S = Set, U = Undefined; *-n* = Value after reset															

Table 7. AD Control Register 1 (ADCR1) Field Descriptions

Bit	Name	Value	Description
15	COS		Continue on suspend enable. This bit affects *emulation operation only*. It defines whether the ADC core clock (ADCLK) is immediately halted when the emulation system enters suspend mode or if it should continue to the end of the converted channel before halting. If COS = 0 when suspend mode is entered, then the accuracy of the conversion results may be affected, depending on the length of the suspension.
		0	Halt conversion immediately when suspend mode is entered.
		1	Continue channel conversion (to completion) when emulation suspend mode is entered.
14	Reserved		Reads are indeterminate. Writes have no effect.

Table 7. AD Control Register 1 (ADCR1) Field Descriptions

Bit	Name	Value	Description
13	CAL EN		Calibration enable. When set to 1, this bit disables the input channel multiplexer, connects the calibration reference selected by HILO and BRIDGE EN to the ADC core input, and starts a calibration conversion when the CAL ST is set to 1. If the CAL ST bit is already to 1 when CAL EN is set to 1, then the conversion is immediately started. Do not set the bit CAL EN to 1 with SELF TST = 1.
		0	*Read:* Calibration is disabled.
		1	*Read:* Calibration is enabled and starts when CAL ST is set.
12	CAL ST		Calibration conversion start. This bit is set-only. Writing a 0 has no effect. Only the end of the conversion signal generated by the ADC core can clear this bit. Setting the CAL ST bit while the CAL EN bit is set starts conversion of the selected reference voltage.
		0	*Read:* Conversion is complete. *Write:* Has no effect.
			Read: Conversion is in process. *Write:* Start calibration conversion.
11	BRIDGE EN		Bridge enable. When set with the HILO bit, BRIDGE EN allows a reference voltage to be converted in calibration mode. Table 8 lists the encoding for BRIDGE EN and HILO with the associated reference voltage selections. In normal mode (SELF TST = 0 and CAL EN = 0), the HILO bit has no effect.
		0	Apply midpoint reference voltage to ADC core input.
		1	Apply full reference voltage to ADC core input.
10	HILO		Test and reference source selection. When self-test is enabled (SELF TST = 1), this bit defines the test voltage to be combined via a resistor with the selected input pin voltage. In calibration mode (CAL EN = 1), this bit defines the reference source polarity (see Table 8). In normal mode (SELF TST = 0 and CAL EN = 0), the HILO bit has no effect.

Table 7. AD Control Register 1 (ADCR1) Field Descriptions

Bit	Name	Value	Description
9	SELF TST		Self-test enable. When SELF TST = 1, the self-test function is enabled. Either AD_{REFHI} or AD_{REFLO} is connected via a resistor to the selected input channel (depending on the HILO bit). The desired conversion mode is configured in the control register ADCR2. The conversion starts after either of the following: - Writing to the input-select registers (ADISR1 or ADISR2) - An external event for the channels selected in the ADEISR register
		0	Self-test mode is disabled.
		1	Self-test mode is enabled.
8	PWR DN		Power-down mode. This bit stops the ADC internal clocks. To release power-down mode, PPWNOVR must be cleared before the PWR DN bit is cleared. In power-down mode, no ADC registers can be read or written. (See Chapter 1, *System Module*, of the *TMS470 Peripheral Module User Guide, Volume 1*, [literature number SPNU189] for a detailed description of PPWNOVR.)
		0	Normal mode
		1	Power-down mode
7–6	Reserved		Reads are indeterminate. Writes have no effect.
5	ADC EN		Analog-to-digital converter enable. When this bit is cleared to 0, the ADC state machine is in idle state. While the ADC EN bit is cleared, the input-select registers (ADEISR, ADISR1, and ADISR2), and the status register (ADSR) are held at their reset values. When ADC EN = 1, the ADC state machine is released, which allows normal ADC operation to start. **Note: At reset, this bit is cleared to 0 and must be set to 1 to enable the ADC functions.**
		0	Disable ADC for normal operation.
		1	Enable ADC for normal operation.

Table 7. AD Control Register 1 (ADCR1) Field Descriptions

Bit	Name	Value	Description
4–3	ACQ.1:0		Acquisition prescale bits. These bits define the prescaler value for the acquisition (sampling) time. The acquisition time of the voltage at the ADC input is defined by: $t_{d(SH)} = t_{c(ADCLK)} * ACQ.1:0$ (see section 2.2.9, *Compatibility Mode Prescaler and ADC Timing*). The acquisition prescaler bits allow the ADC module to achieve better performance by increasing the acquisition time. Increased acquisition time is required for sources that have greater than a 10-ohm source impedance. Table 9 lists the encoding for the prescale values allowed for ACQ.1:0.
2–0	PS.2:0		Clock prescale bits. This bit defines the prescaler value for the ADC clock (ADCLK) as $t_{c(ADCLK)} = t_{c(ICLK)} *$ PS.2:0.

Table 8. Reference Voltages Bit Encoding

BRIDGE EN	HILO	Calibration Mode (CAL EN = 1, SELF TST = 0) Reference Voltage	Self-Test Mode (CAL EN = 0, SELF TST = 1) Reference Voltage
0	0	$(AD_{REFHI}*R1 + AD_{REFLO}*R2) / (R1 + R2)$	AD_{REFLO} via R1 || R2 connected to Vin
0	1	$(AD_{REFLO}*R1 + AD_{REFHI}*R2) / (R1 + R2)$	AD_{REFHI} via R1 || R2 connected to Vin
1	0	AD_{REFLO}	AD_{REFLO} via R1 || R2 connected to Vin
1	1	AD_{REFHI}	AD_{REFHI} via R1 || R2 connected to Vin

Table 9. Acquisition Prescale Value Encoding

Acquisition Prescale Bits		Acquisition Prescale Value (A_{CQ})
ACQ1	**ACQ0**	**ADCLK Cycles**
0	0	2
0	1	8
1	0	32
1	1	128

Note: ADCLK Determined by PS.2:0 * ICLK

The acquisition time is also affected by the clock prescaler value (P_S) described in Table 10.

Table 10. Clock Prescale Bits

ADCLK Clock Prescale Bits			ADCLK Clock Prescale Value (P_S),
PS2	**PS1**	**PS0**	**ICLK Cycles**
0	0	0	1
0	0	1	2
0	1	0	3
0	1	1	4
1	0	0	5
1	0	1	6
1	1	0	7
1	1	1	8

11.2 AD Control Register 2 (ADCR2)

This register is available in compatibility and buffered modes. Figure 16 and Table 11 describe this register.

Figure 16. AD Control Register 2 (ADCR2)

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x04	Reserved				EV MODE	FRZ EV	ENA EV INT	EV EDG SEL	Reserved		G1 MODE	FRZ P1	ENA G1 INT	G2 MODE	FRZ G2	ENA G2 INT
	U				RW-0	RW-0	RW-0	RW-0	U		RW-0	RW-0	RW-0	RW-0	RW-0	RW-0
	R = Read; W = Write; U = Undefined; *-n* = Value after reset															

Table 11. AD Control Register 2 (ADCR2) Field Descriptions

Bit	Name	Value	Description
15–12	Reserved		Reads are indeterminate. Writes have no effect.
11	EV MODE		Event mode. This bit selects whether the event group, as defined by the ADEISR register, is converted in single or continuous conversion mode.
		0	Enable single conversion mode of event group.
		1	Enable continuous conversion mode of event group.

Table 11. AD Control Register 2 (ADCR2) Field Descriptions

Bit	Name	Value	Description
10	FRZ EV		Freeze conversion event group. This bit allows event group conversion flow to be frozen if a group 1 or a group 2 conversion is requested. The conversion for the event group is frozen as long as the group 1 or group 2 conversion is active. When the conversion of group 1 or group 2 is complete, the event group conversion continues from where it was frozen. While this state is maintained, the status bit EV STOP (ADSR.8) is set. When the conversion of group 1 or 2 is complete, the event operation is enabled and resumes from the point at which it was stopped. The ADC clears the EV STOP bit to 0. When the FRZ EV bit is cleared, the current conversion cycle is completed before any new group conversion requests are enabled. On reset, this bit is cleared.
		0	Complete EVT group before enabling another group for conversion.
		1	Freeze EVT group when another group requests conversion.
9	ENA EV INT		ADC enable event group conversion interrupt. This bit enables interrupts at the end of an event group conversion. If this bit is set, an interrupt is requested when the EV END flag in ADSR is set by the sequencer logic. If ENA EV INT is set while EV END = 1, an interrupt is requested immediately.
		0	EVT group interrupt is disabled. No interrupt is sent to the CIM. See Chapter 1, *System Module* (literature number SPNU189), of the *TMS470 Peripheral Module User Guide, Volume 1* for more details.
		1	EVT group interrupt is enabled.

Table 11. AD Control Register 2 (ADCR2) Field Descriptions

Bit	Name	Value	Description
8	EV EDG SEL		ADC event edge select. This bit determines the transition (low-to-high or high-to-low) on the ADEVT pin that triggers event group conversion.
		0	Trigger on high-to-low transition.
		1	Trigger on low-to-high transition.
7–6	Reserved		Reads are indeterminate. Writes have no effect.
5	G1 MODE		Group 1 selection mode. This bit selects whether group 1, defined by the ADISR1 register, is converted in single or continuous conversion mode.
		0	Enable single conversion mode of group 1.
		1	Enable continuous conversion mode of group 1.
4	FRZ G1		FRZ G1**.** Freeze conversion group 1. This bit freezes a group 1 conversion after the current channel is complete if a group 2 or an event group conversion is requested. The conversion for group 1 is frozen as long as the group 2 or event conversion is active. While this state is maintained, the status bit G1 STOP (ADSR.10) is set. When the conversion of the higher priority group (group 2 or group event) is complete, the ADC clears the G1 STOP bit and group 1 operation resumes from the point at which it was frozen. When FRZ G1 = 0, the whole G1 conversion sequence (all channels selected by ADISR1) is completed before any new group conversion requests are enabled.
		0	Complete G1 before enabling another group for conversion.
		1	Freeze G1 when another group requests conversion.

Table 11. AD Control Register 2 (ADCR2) Field Descriptions

Bit	Name	Value	Description
3	ENA G1 INT		Enable group 1 interrupt. This bit enables interrupts at the end of a group 1 conversion. If this bit is set to 1, an interrupt is requested when the G1 END flag in ADSR is set by the sequencer logic. If ENA G1 INT is set while G1 END = 1, an interrupt is requested immediately.
		0	Group 1 interrupt is disabled. No interrupt is sent to the CIM. See Chapter 1, *System Module*, of the *TMS470 Peripheral Module User Guide, Volume 1* (literature number SPNU189) for more details.
		1	Group 1 interrupt is enabled.
2	G2 MODE		Group 2 selection mode. This bit selects whether group 2, defined by the ADISR2 register, is converted in single or continuous conversion mode.
		0	Enable single conversion mode of group 2.
		1	Enable continuous conversion mode of group 2.
			Note: The three groups (G1, G2, and EVT) cannot operate together in continuous mode. When this situation occurs, group 2 is automatically reset to single-conversion mode, and the bit G2 MODE (ADCR2.2) is cleared to reflect the actual conversion mode of group 2.

Table 11. AD Control Register 2 (ADCR2) Field Descriptions

Bit	Name	Value	Description
1	FRZ G1		Freeze conversion group 2. This bit freezes a group 2 conversion after conversion of the current channel is complete if a group 1 or an event group conversion is requested. The conversion for group 2 is frozen as long as the group 1 or event group conversion is active. While this state is maintained, the status bit G2 STOP (ADSR.9) is set. When the conversion of the higher priority group (group 1 or event) is complete, the ADC clears the G2 STOP bit and group 2 operation resumes from the point at which it was frozen. When FRZ G2 = 0, the whole G2 conversion sequence (all channels selected by ADISR2) is completed before any new group conversion requests are enabled.
		0	Complete group 2 before enabling another group for conversion.
		1	Freeze group 2 when another group requests conversion.
0	ENA G2 INT		Enable group 2 interrupt. This bit enables interrupts at the end of a group 2 conversion. If this bit is set to 1, an interrupt is requested when the G2 END flag in ADSR is set by the sequencer logic. If ENA G2 INT is set while G2 END = 1, an interrupt is requested immediately. **Note: This interrupt enable is intended for use in compatibility mode only. In buffered mode, each FIFO has multiple interrupt sources that are enabled in ADBCR3 (see section 11.20, *AD Buffer Control Register 3 (ADBCR3)*). However, this bit is functional in buffered mode; although it is recommended to use the FIFO-based interrupts controlled by ADBCR3.**
		0	Group 2 interrupt is disabled. No interrupt is sent to the CIM. See Chapter 1, *System Module* (literature number SPNU189), of the *TMS470 Peripheral Module User Guide, Volume 1* for more details.
		1	Group 2 interrupt is enabled.

11.3 AD Status Register (ADSR)

This register is available in compatibility and buffered modes. Figure 16 and Table 11 describe this register.

Figure 17. AD Status Register (ADSR)

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x08	Reserved		G1 BUSY	G2 BUSY	EV BUSY	G1 STOP	G2 STOP	EV STOP	Reserved					G1 END	G2 END	EV END
	R-0		R-0	R-0	R-0	R-0	R-0	R-0	R-0					RW-0	RW-0	RW-0
	R = Read; W = Write; U = Undefined; *-n* = Value after reset															

Table 12. AD Status Register (ADSR) Field Descriptions

Bit	Name	Value	Description
15–14	Reserved		These bits always read 0. Writes have no effect.
13	G1 BUSY		Group 1 conversion-busy flag. This bit indicates that a conversion of group 1 is currently active. The sequence can be in a frozen or converting state. While in continuous mode, this bit is read as a 1.
		0	*Read:* Group 1 is not active and can be used for a conversion request. *Write:* Has no effect.
		1	*Read:* Group 1 is already busy. *Write:* Has no effect.

Table 12. AD Status Register (ADSR) Field Descriptions

Bit	Name	Value	Description
12	G2 BUSY		Group 2 conversion-busy flag. This bit indicates that group 2 is currently active. The sequence can be in a frozen or converting state. While in continuous mode, this bit is read as a 1.
		0	*Read:* Group 2 is not active and can be used for a conversion request. *Write:* Has no effect
		1	*Read:* Group 2 is already busy. *Write:* Has no effect.
11	EV BUSY		Event group conversion-busy flag. Indicates that the event group is currently active; the sequence can be in a frozen or converting state. While in continuous mode, this bit is read as a 1.
		0	*Read:* Group event is not active and can be used for a conversion request. *Write:* Has no effect.
		1	*Read:* Group event is already busy. *Write:* Has no effect.
10	G1 STOP		Group 1 conversion-stopped flag. This bit indicates that a group 1 conversion is currently frozen due to arbitration by a group 2 or an event group conversion.
		0	*Read:* Group 1 is not frozen. *Write:* Has no effect.
		1	*Read:* Group 1 is currently frozen. *Write:* Has no effect.
9	G2 STOP		Group 2 conversion-stopped flag. This bit indicates that a group 2 conversion is currently frozen because of arbitration by a group 1 or an event group conversion.
		0	*Read:* Group 2 is not frozen. *Write:* Has no effect.
		1	*Read:* Group 2 is currently frozen. *Write:* Has no effect.

Table 12. AD Status Register (ADSR) Field Descriptions

Bit	Name	Value	Description
8	EV STOP		Event group conversion-stopped flag. This bit indicates that event group conversion is currently frozen due to arbitration by a group 1or group 2 conversion.
		0	*Read:* Group event is not frozen. *Write:* Has no effect.
		1	*Read:* Group event is currently frozen. *Write:* Has no effect.
7–3	Reserved		These bits always read 0. Writes have no effect.
2	G1 END		Group 1 conversion-ended flag. This bit is set when a group 1 conversion is complete (after the A/D conversion of *all* selected channels as specified by ADISR1). This bit is cleared under the following conditions: ❑When reading any ADDRx register associated with a selected channel as specified by ADISR1 ❑When writing to ADISR1 ❑Writing a 1 to G1 END ❑When ADC EN = 0 **Note: If ENA G1 INT is set when G1 END is set, an interrupt is requested.** **Note: ADDR registers can be overwritten. Having the G1 END flag set does not prevent the ADC from writing new conversion results into the configured digital result registers (ADDRx).**
		0	*Read:* Group 1 conversion has not completed since this flag was cleared. *Write:* Has no effect.
		1	*Read:* Group 1 conversion is complete. *Write:* Clears the G1 END flag.

Table 12. AD Status Register (ADSR) Field Descriptions

Bit	Name	Value	Description
1	G2 END		Group 2 conversion-ended flag. This bit is set when a group 2 conversion is complete (after the A/D conversion of *all* of the selected channels as specified by ADISR2). This bit is cleared under the following conditions: ❏ When reading any ADDRx register associated with a selected channel as specified by ADISR2 ❏ When writing to ADISR2 ❏ Writing a 1 to G2 END ❏ When ADC EN = 0 **Note: If ENA G2 INT is set when G2 END is set, an interrupt is requested.** **Note: ADDR registers can be overwritten. Having the G2 END flag set does not prevent the ADC from writing new conversion results into the configured digital result registers (ADDRx).**
		0	*Read:* Group 1 conversion has not completed since this flag was cleared. *Write:* Has no effect.
		1	*Read:* Group 1 conversion is complete. *Write:* Clears the G2 END flag

Table 12. AD Status Register (ADSR) Field Descriptions

Bit	Name	Value	Description
0	EV END		Event conversion-ended flag. This bit is set when the EVT group conversion is complete (after the A/D conversion of *all* of the selected channels as specified by ADEISR). This bit is cleared under the following conditions: ❑When reading any ADDRx register associated with a selected channel as specified by ADEISR ❑When writing to ADEISR ❑Writing a 1 to EV END ❑When ADC EN = 0 **Note: If ENA EV INT is set when G1 END is set, an interrupt is requested.** **Note: ADDR registers can be overwritten. Having the EV END flag set does not prevent the ADC from writing new conversion results into the configured digital result registers (ADDRx).**
		0	*Read:* EVT group conversion has not completed since this flag was cleared. *Write:* Has no effect.
		1	*Read:* EVT group conversion is complete. *Write:* Clears the EV END flag

11.4 AD Event Group Input Channel Select Register (ADEISR)

The event group input channel select register controls the ability of the ADC to start an automatic conversion sequence (from one single channel up to 16 channels) based on an asynchronous event occurrence at the ADEVT pin.

For each bit set in ADEISR, one channel is selected for conversion when the event occurs. For example, if bits 11, 3, and 1 are to be converted, then write the value 0000100000001010 binary (0x080A) to ADEISR. After an event at the ADEVT pin occurs, the event group conversions start. Channel 1 is converted first, then channel 3, then channel 11.

Conversions begin when an edge whose polarity is selected by EV EDGE SEL occurs on the source selected by EVSRC[1:0]. If *n* is the number of the first channel selected in ADEISR, then the event conversion will start *n*+1 ICLK clock cycles (***not ADCLK***) after the event or immediately after the end of the current conversion cycle, if a group conversion is in progress (depending on the freeze group bit settings of the other two groups).

When more than one channel is selected, the channels are converted in order from lowest to highest.

This register is available in compatibility and buffered modes. Figure 18 and Table 13 describe this register.

Figure 18. AD Event Group Input Channel Select Register (ADEISR)

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0C	EV.15:0															
	RW-0															
	R = Read; W = Write; S = Set; C = Clear; U = Undefined; *-n* = Value after reset															

Table 13. AD Event Group Input Channel Select Register (ADEISR) Field Descriptions

Bit	Name	Value	Description
15–0	EV.15:0		A/D Event channel 15 to 0 selection bits. When this bit is set, the corresponding channels are enabled for subsequent conversion after an external event. The result of each conversion is stored in its corresponding data register (ADDR0 to ADDR15). Writing 0x0000 to ADEISR stops event group conversions. **Note: Writing a new value to ADEISR while a current event group is being converted does not reset the ongoing channel scan process. This action results in a new sequence starting from the last channel converted in the previous ADEISR selection.**
		0	The ADC input channel will not be converted in this group.
		1	The ADC input channel will be converted in this group.

11.5 AD Group 1 Input Channel Select Register (ADISR1)

The input channel select register 1 controls the ability of the ADC to start a sequential conversion of selected channels of group 1 (from a single channel up to 16 channels).

For each bit set in ADISR1, one channel is selected for conversion when ADISR1 is written. For example, if bits 11, 3, and 1 are to be converted, then when the value 0000100000001010 binary (0x080A) is written to ADISR1, group 1 conversions start. Channel 1 is converted first, then channel 3, then channel 11.

While another group is being converted (event or group 2), the group 1 conversion is delayed if the freeze bit of the active group = 0. If the freeze bit of the active group = 1, then that group is frozen as soon as ADISR1 is written, and group 1 conversion starts after the conversion of the current channel of the active group is complete. When the group 1 conversion is complete, the conversion of the previously active group resumes where it left off.

When more than one channel is selected, the channels are converted in order from lowest to highest.

This register is available in compatibility and buffered modes. Figure 19 and Table 14 describe this register.

Figure 19. AD Group 1 Input Channel Select Register (ADISR1)

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x10	G1.15:0															
	RW-0															
	R = Read; W = Write; S = Set; C = Clear; U = Undefined; *-n* = Value after reset															

Table 14. AD Group 1 Input Channel Select Register (ADISR1) Field Descriptions

Bit	Name	Value	Description
15–0	G1.15:0		A/D channel 15 to 0 enable bits. When this bit is set, the corresponding channels are enabled for subsequent conversion according to the group 1 configuration. Each channel's resulting conversion is stored into a dedicated data register (ADDR0 to ADDR15). Any write with a value other than 0x0000 starts a group 1 operation. Writing a 0x0000 value stops group 1 operation. **Note: The conversion sequence is always from lowest to highest. No conversion takes place for a channel not enabled, and the corresponding data register remains at its previous value. When a channel is disabled, it takes only 1 ICLK cycle to skip over it to the next higher channel.**
		0	The ADC input channel will not be converted in this group.
		1	The ADC input channel will be converted in this group.

11.6 AD Group 2 Input Channel Select Register (ADISR2)

The input channel select register 2 controls the ability of the ADC to start a sequential conversion of the selected channels (from a single channel up to 16 channels).

For each bit set in ADISR2, one channel is selected for conversion when ADISR2 is written. For example, if bits 11, 3, and 1 are to be converted, then when the value 0000100000001010 binary (0x080A) is written to ADISR2, group 2 conversions start. Channel 1 is converted first, then channel 3, then channel 11.

While another group is being converted (event or group 1), the group 2 conversion is delayed if the freeze bit of the active group = 0. If the freeze bit of the active group = 1, then that group is frozen as soon as ADISR2 is written, and conversion of group 2 starts after the completion of the active group's current channel conversion. When the group 2 conversion is complete, the previously active group resumes where it left off.

When more than one channel is selected, the channels are converted in order from lowest to highest.

This register is available in compatibility and buffered modes. Figure 20 and Table 15 describe this register.

Figure 20. AD Group 2 Input Channel Select Register (ADISR2)

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x14	G2.15:0															
	RW-0															
	R = Read; W = Write; S = Set; C = Clear; U = Undefined; *-n* = Value after reset															

Table 15. AD Group 2 Input Channel Select Register (ADISR2) Field Descriptions

Bit	Name	Value	Description
15–0	G2.15:0		A/D channel 15–0 enable bits. When this bit is set, the corresponding channels are enabled for subsequent conversion according to the group 2 configuration. Each channel's resulting conversion is stored into a dedicated data register (ADDR0 to ADDR15). Any write with a value other than 0x0000 starts a group 2 operation. Writing a 0x0000 value stops group 2 operation. **Note: The conversion sequence is always from lowest to highest. No conversion takes place for a channel not enabled, and the corresponding data register remains at its previous value. When a channel is disabled, it takes only 1 ICLK cycle to skip over it to the next higher channel.**
		0	The ADC input channel will not be converted in this group.
		1	The ADC input channel will be converted in this group.

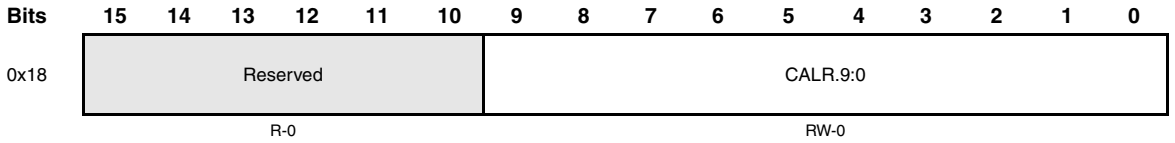
11.7 Calibration and Offset Error Correction Register (ADCALR)

The ADCALR register receives data from the ADC core after a calibration conversion while calibration mode is enabled (CAL EN = 1). To calibrate the ADC, the calibration data must be read from the ADCALR register by the CPU during calibration mode. Then an offset error correction value must be calculated by the CPU. The CPU must then write the offset error correction value back into ADCALR in 2s complement form before corrections can begin.

During normal conversion (when calibration is disabled), the register content is automatically added to each digital result output from the ADC core before it is stored in one of the digital result registers (ADDR0 to ADDR15).

This register is available in compatibility and buffered modes. Figure 21 and Table 16 describe this register.

Figure 21. Calibration and Offset Error Correction Register (ADCALR)



R = Read; W = Write; S = Set; C = Clear; U = Undefined; *-n* = Value after reset

Table 16. Calibration and Offset Error Correction Register (ADCALR) Field Descriptions

Bit	Name	Value	Description
15–10	Reserved		These bits always read 0. Writes have no effect.
9–0	CALR.9:0		Calibration bits. These 10 bits are cleared to 0 after a system reset or by clearing the register to 0. They are not cleared when ADC EN (ADCR1.5) is written to 0. The ADC loads these bits in calibration mode; the application program computes the calibrated value and loads the value in noncalibration mode. ❑In calibration mode, the conversion result corresponding to the selected calibration reference, as determined by BRIDGE EN and HILO, is written to CALR.9:0 where it can be read by the application program. When all the calibration data is collected, the application program can write a 2s complement offset error correction value to this register. A value of 0x000 leaves the conversion results unchanged when it is written to the digital result register frame (ADDR0 to ADDR15). These bits can be written by the CPU in calibration mode; however, they will be overwritten by any subsequent conversion. ❑In noncalibration mode, the application should write the error offset value to CALR.9:0 where it is subsequently added to each conversion result automatically by the ADC before the conversion is stored in the digital result register frame. On system reset, ADCALR is cleared. Write: 2s complement form of application-calculated error offset value during noncalibration mode Read: Calibration data in calibration mode, or offset error data in noncalibration mode **Note: ADCALR can be written out of calibration mode. Any value written to this register in noncalibration mode modifies the result from the ADC core. For no correction, write 0x000 to ADCALR. The correction value must be stored in ADCALR in 2s complement form.**

11.8 Digital Result Registers (ADDR15–ADDR0)

Each of these 16 registers contains the status flag and digital results of their associated channel x, where $15 \geq x \geq 0$. A read of a digital result register sets the corresponding DTxST bit to 1 when the read is completed and clears the G1 END, G2 END, or EV END flags, depending on the channel versus group selection.

These registers are available in compatibility mode only. Figure 22 and Table 17 describe these registers.

Figure 22. Digital Result Registers (ADDR15–ADDR0)

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x1C to 0x90	DTx ST	Reserved					DTx.9:0									
	R-X	R-0					R-X									

R = Read; W = Write; S = Set; C = Clear; U = Undefined; -X = Indeterminate after reset; *-n* = Value after reset

Table 17. Digital Result Registers (ADDR15–ADDR0) Field Descriptions

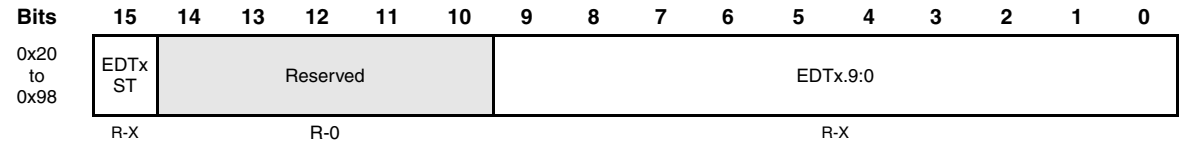
Bit	Name	Value	Description
15	DtxST		Data read status. This bit indicates that the last digital result loaded into DTx.9:0 has already been read by the CPU. It is cleared when the converter loads a new conversion result, and is set *after* the CPU reads the digital result. After reset, the value is indeterminate.
		0	*Read:* The data DTx.9:0 has not been read by the CPU since the last update. *Write:* Has no effect.
		1	*Read:* The data DTx.9:0 has already been read by the CPU since the last update. *Write:* Has no effect.
14–10	Reserved		These bits always read 0. Writes have no effect.
9–0	DTx.9:0		Conversion data bits 9 to 0. This bit stores the IO bit results of the corresponding conversion of the associated analog input channel. Write:Has no effect Read:Clears the DTxST flag and appropriate END flag

11.9 Emulation Digital Result Registers (ADEMDR15-ADEMDR0)

These 16 registers support *emulation* and are a duplication of registers ADDR15 to ADDR0 except that reading ADEMDRx *does not* affect the DTx ST bit in the ADDRx register nor does it clear the G1 END, G2 END, or EV END flags. This allows emulation software to read the register without affecting its contents or causing any other actions to take place.

These registers are available in compatibility mode only. Figure 23 and Table 18 describe these registers.

Figure 23. Emulation Digital Result Registers (ADEMDR15–ADEMDR0)



R = Read; W = Write; S = Set; C = Clear; U = Undefined; -X = Indeterminate after reset; -*n* = Value after reset

Table 18. Emulation Digital Result Registers (ADEMDR15–ADEMDR0) Field Descriptions

Bit	Name	Value	Description
15	EDTxST		EDTxST. Data read status. This bit reflects the current state of the DTx ST bit in the ADDRx register, but *is not* affected when ADEMDRx is read. After reset, the value is indeterminate.
		0	*Read:* The data DTx.9:0 has not been read by the CPU since the last update. *Write:* Has no effect.
		1	*Read:* The data DTx.9:0 has already been read by the CPU since the last update. *Write:* Has no effect.
14–10	Reserved		These bits always read 0. Writes have no effect.
9–0	EDTx.9:0.		Conversion data bits 9 to 0. This bit reflects the current state of DTx.9:0 in the ADDRx register.

11.10 Data Input Register (ADINR)

For each input pin, the register ADINR contains an associated bit to indicate whether the input voltage was lower or higher than the midpoint of the reference voltage at the time that channel was last converted.

The register bits IN15:0 are copies of the most significant bits of the conversion results residing in bit positions ADDR15.9 through ADDR0.9.

The ADINR register acts as a digital input register and can be read by the application program to determine the digital level at the input pins ADIN15 to ADIN0. This data is only as valid as the latest conversion data. If a channel has not been converted, then the corresponding bit in ADIN is not valid.

This register is available in compatibility mode and buffered mode. Figure 24 and Table 19 describe this register.

Figure 24. Data Input Register (ADINR)

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x9C	IN15:0															
	R-X															

R = Read; W = Write; S = Set; C = Clear; U = Undefined; -X = Indeterminate after reset; -*n* = Value after reset

Table 19. Data Input Register (ADINR) Field Descriptions

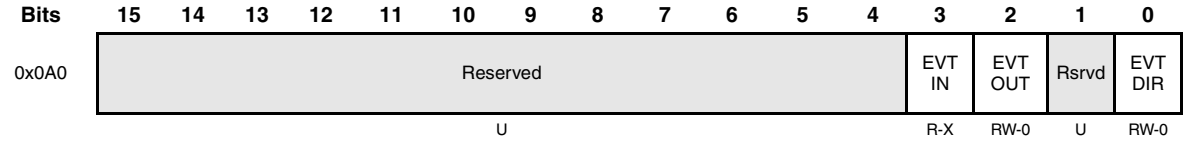
Bit	Name	Value	Description
15–0	IN15:0		Digital input pins. This bit stores the digital level of each of the 16 analog input pins as determined by the latest conversion result. After reset, the value is indeterminate.
		0	*Read:* A level lower than the midpoint reference voltage was measured at the last conversion. *Write:* Has no effect.
		1	*Read:* A level higher than midpoint reference voltage was measured at the last conversion. *Write:* Has no effect.

11.11 Pin Control Register (ADPCR)

The ADPCR register contains the digital value (high or low) of the current level at the ADEVT pin. It also defines the direction of the ADEVT pin and the data to be output to this pin. This pin can be used as a trigger for an event-controlled conversion or as a general-purpose I/O.

This register is available in compatibility mode only. Figure 25 and Table 20 describe this register.

Figure 25. Pin Control Register (ADPCR)



R = Read; W = Write; S = Set; C = Clear; U = Undefined; -X = Indeterminate after reset; *-n* = Value after reset

Table 20. Pin Control Register (ADPCR) Field Descriptions

Bit	Name	Value	Description
15–4	Reserved		Reads are indeterminate. Writes have no effect.
3	EVT IN		ADEVT pin input value. This bit reflects the logical level at the ADEVT pin, *regardless* of the direction bit. After reset, the value is indeterminate. It reflects the state of the ADEVT pin.
		0	*Read:* Logic low on the ADEVT pin (input voltage is V_{IL} or lower) *Write:* Has no effect.
		1	*Read:* Logic high on the ADEVT pin (input voltage is V_{IH} or higher) *Write:* Has no effect.

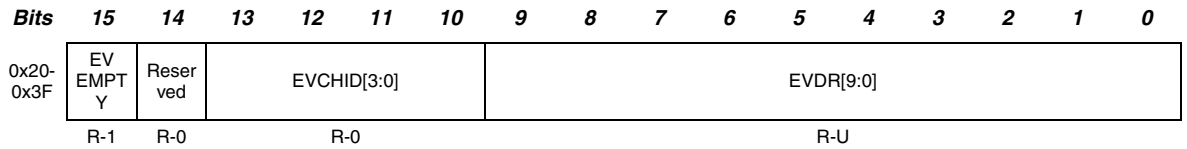
Table 20. Pin Control Register (ADPCR) Field Descriptions (Continued)

Bit	Name	Value	Description
2	EVT OUT		ADEVT pin data output. This bit determines the data to be output to the ADEVT pin when the direction bit, EVT DIR, is set.
		0	Logic low on the ADEVT pin (output voltage is V_{OL} or lower)
		1	Logic high on the ADEVT pin (output voltage is V_{OH} or higher)
			Note: ADEVT may be used as a software trigger. If the input-select register (ADEISR) is not 0x0000, then toggling the ADEVT pin (either from an external source or by writing to the ADPCR register) starts an event group conversion. For this conversion to happen, the edge selection bit, EV EDG SEL, must agree with the direction of the toggle. The conversion is disabled only when ADEISR = 0x0000. See Section 2.2.4, *Event Edge Detector*, on page 12, Section 11.2, *AD Control Register 2 (ADCR2)*, on page 56, and Section 11.4, *AD Event Group Input Channel Select Register (ADEISR)*, on page 66 for more details.
1	Reserved		Reads are indeterminate. Writes have no effect.
0	EVT DIR		ADEVT pin direction selection. This bit determines whether the ADEVT pin is an input or an output pin.
		0	ADEVT output buffer is disabled.
		1	ADEVT output buffer is enabled.

11.12 AD Event FIFO Buffer (ADBUFE)

This register is available in buffered mode only. Figure 26 and Table 21 describe this register.

Figure 26. AD Event FIFO Buffer (ADBUFE)



R = Read, W = Write, S = Set, U = Undefined, *-n* = Value after reset

Table 21. AD Event FIFO Buffer (ADBUFE) Field Descriptions

Bit	Name	Value	Description
15	EV EMPTY		Event Group FIFO Empty
		0	The data is valid.
		1	The FIFO buffer is empty and the FIFO data is meaningless.
14	Reserved		Reads 0 always. Writes have no effect.
13–10	EVCHID[3:0]		These bits read 0 unless the CHID bit in ADBUFCR2 is enabled. When enabled, these bits identify the A/D channel from which the conversion result stored in EVDR[9:0] was obtained.
		0000	The conversion result is from A/D channel 0, or the channel ID mode is disabled in ADBUFCR2.
		0001	The conversion result is from A/D channel 1.
		0010	The conversion result is from A/D channel 2.
		0011	The conversion result is from A/D channel 3.
		0100	The conversion result is from A/D channel 4.
		0101	The conversion result is from A/D channel 5.
		0110	The conversion result is from A/D channel 6.
		0111	The conversion result is from A/D channel 7.

Table 21. AD Event FIFO Buffer (ADBUFE) Field Descriptions (Continued)

Bit	Name	Value	Description
		1000	The conversion result is from A/D channel 8.
		1001	The conversion result is from A/D channel 9.
		1010	The conversion result is from A/D channel 10.
		1111	The conversion result is from A/D channel 11.
		1100	The conversion result is from A/D channel 12.
		1101	The conversion result is from A/D channel 13.
		1110	The conversion result is from A/D channel 14.
		1111	The conversion result is from A/D channel 15.
9–0	EVDR9:0		Event Group Digital Result. These bits contain the digital result output from the Event Group FIFO buffer. The result can be presented in two formats: 8 bit and 10 bit. By default, the result is presented in 10-bit format. If the EV8BIT bit is set in the ADBUFCR2 register, then EVDR[9:8] read 00 and EVDR[7:0] read the 8 MSBs of the 10-bit digital result (that is, EVDR[7:0] read the value normally in EVDR[9:2] when EV8BIT is disabled). Note that the data shifting is performed during read out of the FIFO so that the EV8BIT mode can be changed dynamically. **Note: Reading this register updates the event group FIFO buffer OUT pointer if the read is valid (that is, if the FIFO buffer is not EMPTY or OVERRUN).** **Note: The FIFO buffer does NOT update with SUSPEND active. The FIFO buffer is not updated under any circumstances when this register is read if SUSPEND is active.** **Note: This register is aliased eight times. Any word-aligned read within the range 0x20– 0x3F results in the EVENT group FIFO being read. This allows the LDMIA instruction of the CPU to read out up to eight data values from the FIFO with one instruction.**

11.13 AD EMU Event FIFO Buffer (ADEMBUFE)

This register is available in buffered mode only. Figure 27 and Table 22 describe this register.

Figure 27. AD EMU Event FIFO Buffer (ADEMBUFE)

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x90	EV EMPTY	0	EVCHID[3:0]				EVDR[9:0]									
	R-1	R-0	R-0				R-U									

R = Read, W = Write, S = Set, U = Undefined, *-n* = Value after reset

Table 22. AD EMU Event FIFO Buffer (ADEMBUFE) Field Descriptions

Bit	Name	Value	Description
15	EV EMPTY		Event Group FIFO Empty
		0	The data is valid.
		1	The FIFO buffer is empty and the FIFO data is meaningless.
14	Reserved		Reads 0 always. Writes have no effect.
13–10	EVCHID[3:0]		These bits read 0 unless the CHID bit in ADBUFCR2 is enabled. When enabled, these bits identify the A/D channel from which the conversion result stored in EVDR[9:0] was obtained.
		0000	The conversion result is from A/D channel 0, or the channel ID mode is disabled in ADBUFCR2.
		0001	The conversion result is from A/D channel 1.
		0010	The conversion result is from A/D channel 2.
		0011	The conversion result is from A/D channel 3.
		0100	The conversion result is from A/D channel 4.
		0101	The conversion result is from A/D channel 5.
		0110	The conversion result is from A/D channel 6.
		0111	The conversion result is from A/D channel 7.

Table 22. AD EMU Event FIFO Buffer (ADEMBUFE) Field Descriptions (Continued)

Bit	Name	Value	Description
		1000	The conversion result is from A/D channel 8.
		1001	The conversion result is from A/D channel 9.
		1010	The conversion result is from A/D channel 10.
		1111	The conversion result is from A/D channel 11.
		1100	The conversion result is from A/D channel 12.
		1101	The conversion result is from A/D channel 13.
		1110	The conversion result is from A/D channel 14.
		1111	The conversion result is from A/D channel 15.
9–0	EVDR9:0		Event Group Digital Result. These bits contain the digital result output from the Event Group FIFO buffer. The result can be presented in two formats: 8 bit and 10 bit. By default, the result is presented in 10-bit format. If the EV8BIT bit is set in the ADBUFCR2 register, then EVDR[9:8] read 00 and EVDR[7:0] read the 8 MSBs of the 10-bit digital result (that is, EVDR[7:0] read the value normally in EVDR[9:2] when EV8BIT is disabled). Note that the data shifting is performed during read out of the FIFO so that the EV8BIT mode can be changed dynamically. **Note: FIFO buffer OUT pointer is NOT updated. Reading this register does not update the event group FIFO buffer OUT pointer. This register is used for debugging monitors.**

11.14 AD Group 1 FIFO Buffer (ADBUF1)

This register is available in buffered mode only. Figure 28 and Table 23 describe this register.

Figure 28. AD Group 1 FIFO Buffer (ADBUF1)

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x40-0x5F	G1 EMPTY	Rsrvd	G1CHID				G1DR[9:0]									
	R-1	R-0	R-0				R-U									

R = Read, W = Write, S = Set, U = Undefined, *-n* = Value after reset

Table 23. AD Group 1 FIFO Buffer (ADBUF1) Field Descriptions

Bit	Name	Value	Description
15	G1 EMPTY		Group 1 FIFO Empty
		0	The data is valid.
		1	The FIFO buffer is empty and the FIFO data is meaningless.
14	Reserved		Reads 0 always. Writes have no effect.
13–10	G1CHID[3:0]		These bits read 0 unless the CHID bit in ADBUFCR2 is enabled. When enabled, these bits identify the A/D channel from which the conversion result stored in EVDR[9:0] was obtained.
		0000	The conversion result is from A/D channel 0, or the channel ID mode is disabled in ADBUFCR2.
		0001	The conversion result is from A/D channel 1.
		0010	The conversion result is from A/D channel 2.
		0011	The conversion result is from A/D channel 3.
		0100	The conversion result is from A/D channel 4.
		0101	The conversion result is from A/D channel 5.
		0110	The conversion result is from A/D channel 6.
		0111	The conversion result is from A/D channel 7.

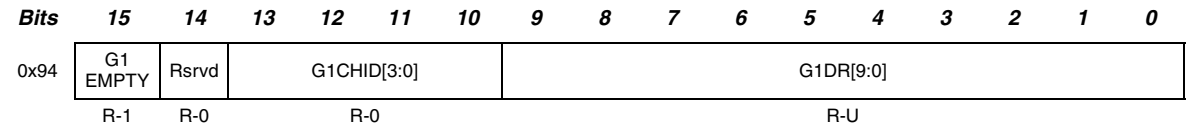
Table 23. AD Group 1 FIFO Buffer (ADBUF1) Field Descriptions (Continued)

Bit	Name	Value	Description
		1000	The conversion result is from A/D channel 8.
		1001	The conversion result is from A/D channel 9.
		1010	The conversion result is from A/D channel 10.
		1111	The conversion result is from A/D channel 11.
		1100	The conversion result is from A/D channel 12.
		1101	The conversion result is from A/D channel 13.
		1110	The conversion result is from A/D channel 14.
		1111	The conversion result is from A/D channel 15.
9–0	G1DR9:0		Group 1 Digital Result. These bits contain the digital result output from the Group 1 FIFO buffer. The result can be presented in two formats: 8 bit and 10 bit. By default, the result is presented in 10-bit format. If the G18BIT bit is set in the ADBUFCR2 register, then G1DR[9:8] read 00 and G1DR[7:0] read the 8 MSBs of the 10-bit digital result (that is, G1DR[7:0] read the value normally in G1DR[9:2] when the G18BIT is disabled). Note that the data shifting is performed during read out of the FIFO so that the G18BIT mode can be changed dynamically. **Note: Update FIFO buffer OUT pointer. Reading this register updates the Group 1 FIFO buffer OUT pointer if the read is valid (that is, if the FIFO buffer is not EMPTY or OVERRUN).** **Note: The FIFO buffer does NOT update with SUSPEND active. The FIFO buffer is not updated under any circumstances when this register is read if SUSPEND is active.** **Note: This register is aliased eight times. Any word-aligned read within the range 0x20– 0x3F results in the EVENT group FIFO being read. This allows the LDMIA instruction of the CPU to read out up to eight data values from the FIFO with one instruction.**

11.15 AD EMU Group 1 FIFO Buffer (ADEMBUF1)

This register is available in buffered mode only. Figure 29 and Table 23 describe this register.

Figure 29. AD EMU Group 1 FIFO Buffer (ADEMBUF1)



R = Read, W = Write, S = Set, U = Undefined, *-n* = Value after reset

Table 24. AD EMU Group 1 FIFO Buffer (ADEMBUF1) Field Descriptions

Bit	Name	Value	Description
15	G1 EMPTY		Group 1 FIFO Empty
		0	The data is valid.
		1	The FIFO buffer is empty and the FIFO data is meaningless.
14	Reserved		Reads 0 always. Writes have no effect.
13–10	G1CHID[3:0]		These bits read 0 unless the CHID bit in ADBUFCR2 is enabled. When enabled, these bits identify the A/D channel from which the conversion result stored in G1DR[9:0] was obtained.
		0000	The conversion result is from A/D channel 0, or the channel ID mode is disabled in ADBUFCR2.
		0001	The conversion result is from A/D channel 1.
		0010	The conversion result is from A/D channel 2.
		0011	The conversion result is from A/D channel 3.
		0100	The conversion result is from A/D channel 4.
		0101	The conversion result is from A/D channel 5.
		0110	The conversion result is from A/D channel 6.
		0111	The conversion result is from A/D channel 7.

Table 24. AD EMU Group 1 FIFO Buffer (ADEMBUF1) Field Descriptions (Continued)

Bit	Name	Value	Description
		1000	The conversion result is from A/D channel 8.
		1001	The conversion result is from A/D channel 9.
		1010	The conversion result is from A/D channel 10.
		1111	The conversion result is from A/D channel 11.
		1100	The conversion result is from A/D channel 12.
		1101	The conversion result is from A/D channel 13.
		1110	The conversion result is from A/D channel 14.
		1111	The conversion result is from A/D channel 15.
9–0	G1DR9:0		Group 1 Digital Result. These bits contain the digital result output from the Group 1 FIFO buffer. The result can be presented in two formats: 8 bit and 10 bit. By default, the result is presented in 10-bit format. If the G18BIT bit is set in the ADBUFCR2 register, then G1DR[9:8] read 00 and G1DR[7:0] read the 8 MSBs of the 10-bit digital result (that is, G1DR[7:0] read the value normally in G1DR[9:2] when the G18BIT is disabled). Note that the data shifting is performed during read out of the FIFO so that the G18BIT mode can be changed dynamically. **Note: FIFO buffer OUT pointer is NOT updated. Reading this register does NOT update the Group 1 FIFO buffer OUT pointer. This register is used for debugging monitors.**

11.16 AD Group 2 FIFO Buffer (ADBUF2)

This register is available in buffered mode only. Figure 30 and Table 25 describe this register.

Figure 30. AD Group 2 FIFO Buffer (ADBUF2)

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x60-0x7F	G2 EMPTY	0	G2CHID[3:0]				G2DR[9:0]									
	R-1	R-0	R-0				R-U									

R = Read, W = Write, S = Set, U = Undefined, *-n* = Value after reset

Table 25. AD Group 2 FIFO Buffer (ADBUF2) Field Descriptions

Bit	Name	Value	Description
15	G2 EMPTY		Group 2 FIFO Empty
		0	The data is valid.
		1	The FIFO buffer is empty and the FIFO data is meaningless.
14	Reserved		Reads 0 always. Writes have no effect.
13–10	G2CHID[3:0]		These bits read 0 unless the CHID bit in ADBUFCR2 is enabled. When enabled, these bits identify the A/D channel from which the conversion result stored in G2DR[9:0] was obtained.
		0000	The conversion result is from A/D channel 0, or the channel ID mode is disabled in ADBUFCR2.
		0001	The conversion result is from A/D channel 1.
		0010	The conversion result is from A/D channel 2.
		0011	The conversion result is from A/D channel 3.
		0100	The conversion result is from A/D channel 4.
		0101	The conversion result is from A/D channel 5.
		0110	The conversion result is from A/D channel 6.
		0111	The conversion result is from A/D channel 7.

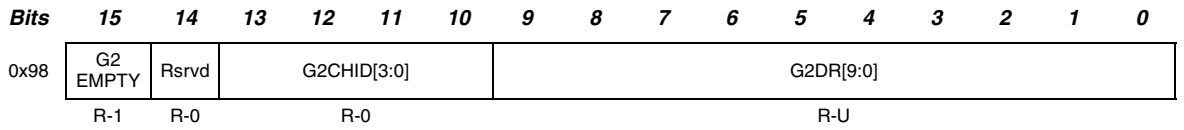
Table 25. AD Group 2 FIFO Buffer (ADBUF2) Field Descriptions (Continued)

Bit	Name	Value	Description
		1000	The conversion result is from A/D channel 8.
		1001	The conversion result is from A/D channel 9.
		1010	The conversion result is from A/D channel 10.
		1111	The conversion result is from A/D channel 11.
		1100	The conversion result is from A/D channel 12.
		1101	The conversion result is from A/D channel 13.
		1110	The conversion result is from A/D channel 14.
		1111	The conversion result is from A/D channel 15.
9–0	G2DR9:0		Group 2 Digital Result. These bits contain the digital result output from the Group 2 FIFO buffer. The result can be presented in two formats: 8 bit and 10 bit. By default, the result is presented in 10-bit format. If the G28BIT bit is set in the ADBUFCR2 register, then G2DR[9:8] read 00 and G2DR[7:0] read the 8 MSBs of the 10-bit digital result (that is, G2DR[7:0] read the value normally in G2DR[9:2] when the G28BIT is disabled). Note that the data shifting is performed during read out of the FIFO so that the G28BIT mode can be changed dynamically. **Note: Update FIFO buffer OUT pointer. Reading this register updates the Group 2 FIFO buffer OUT pointer. This register is used for debugging monitors.** **Note: The FIFO buffer does NOT update with SUSPEND active. The FIFO buffer is not updated under any circumstances when this register is read if SUSPEND is active.** **Note: This register is aliased eight times. Any word-aligned read within the range 0x20– 0x3F results in the Group 2 FIFO being read. This allows the LDMIA instruction of the CPU to read out up to eight data values from the FIFO with one instruction.**

11.17 AD EMU Group 2 FIFO Buffer (ADEMBUF2)

This register is available in buffered mode only. Figure 31 and Table 26 describe this register.

Figure 31. AD EMU Group 2 FIFO Buffer (ADEMBUF2)



R = Read, W = Write, S = Set, U = Undefined, *-n* = Value after reset

Table 26. AD EMU Group 2 FIFO Buffer (ADEMBUF2) Field Descriptions

Bit	Name	Value	Description
15	G2 EMPTY		Group 2 FIFO Empty
		0	The data is valid.
		1	The FIFO buffer is empty and the FIFO data is meaningless.
14	Reserved		Reads 0 always. Writes have no effect.
13–10	G2CHID[3:0]		These bits read 0 unless the CHID bit in ADBUFCR2 is enabled. When enabled, these bits identify the A/D channel from which the conversion result stored in G2DR[9:0] was obtained.
		0000	The conversion result is from A/D channel 0, or the channel ID mode is disabled in ADBUFCR2.
		0001	The conversion result is from A/D channel 1.
		0010	The conversion result is from A/D channel 2.
		0011	The conversion result is from A/D channel 3.
		0100	The conversion result is from A/D channel 4.
		0101	The conversion result is from A/D channel 5.
		0110	The conversion result is from A/D channel 6.
		0111	The conversion result is from A/D channel 7.

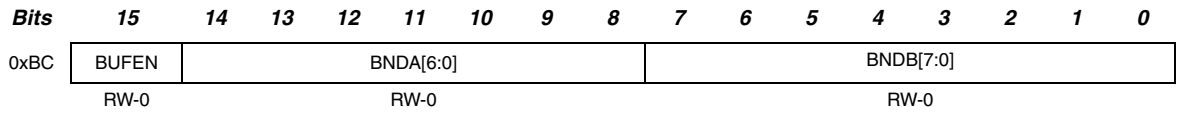
Table 26. AD EMU Group 2 FIFO Buffer (ADEMBUF2) Field Descriptions (Continued)

Bit	Name	Value	Description
		1000	The conversion result is from A/D channel 8.
		1001	The conversion result is from A/D channel 9.
		1010	The conversion result is from A/D channel 10.
		1111	The conversion result is from A/D channel 11.
		1100	The conversion result is from A/D channel 12.
		1101	The conversion result is from A/D channel 13.
		1110	The conversion result is from A/D channel 14.
		1111	The conversion result is from A/D channel 15.
9–0	G2DR9:0		Group 2 Digital Result. These bits contain the digital result output from the Group 2 FIFO buffer. The result can be presented in two formats: 8 bit and 10 bit. By default, the result is presented in 10-bit format. If the G28BIT bit is set in the ADBUFCR2 register, then G2DR[9:8] read 00 and G2DR[7:0] read the 8 MSBs of the 10-bit digital result (that is, G2DR[7:0] reads the value normally in G2DR[9:2] when the G28BIT is disabled). Note that the data shifting is performed during read out of the FIFO so that the G28BIT mode can be changed dynamically. **Note: FIFO buffer OUT pointer is NOT updated. Reading this register does NOT update the Group 1 FIFO buffer OUT pointer. This register is used for debugging monitors.**

11.18 AD Buffer Control Register 1 (ADBCR1)

This register is available in compatibility and buffered mode. Figure 32 and Table 27 describe this register.

Figure 32. AD Buffer Control Register 1 (ADBCR1)



R = Read, W = Write, S = Set, U = Undefined, *-n* = Value after reset

Table 27. AD Buffer Control Register 1 (ADBCR1) Field Descriptions

Bit	Name	Value	Description
15	BUFEN		Buffer Mode Enable
		0	Select compatibility mode.
		1	Select buffered mode.
14–8	BNDA[6:0]		Buffer Boundary A These bits determine the boundary between the two-port RAM space allocated for the EVENT group FIFO, and that of conversion group 1. The boundary is specified in units of two words as an offset from the beginning of the buffer. **Note: *Always* program this register greater than the beginning of the buffer (non-0) and less than BNDB[7:0].**
7–0	BDNB[7:0]		Buffer Boundary B. These bits determine the boundary between the two-port RAM space allocated for the conversion group 1 FIFO and that of conversion group 2. The boundary is specified in units of two words as an offset from the beginning of the buffer. **Note: *Always* program this register greater than BNDA[6:0].**

11.19 AD Buffer Control Register 2 (ADBCR2)

This register is available in compatibility and buffered mode. Figure 33 and Table 28 describe this register.

Figure 33. AD Buffer Control Register 2 (ADBCR2)

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0xC0	RESERVED													BNDEND[2:0]		
	R-U													RW-0		

R = Read, W = Write, S = Set, U = Undefined, *-n* = Value after reset

Table 28. AD Buffer Control Register 2 (ADBCR2) Field Descriptions

Bit	Name	Value	Description
15–3	Reserved		Reads are indeterminate. Writes have no effect.
2–0	BNDEND[2:0]		Buffer End Boundary These bits determine the end boundary of the FIFO buffer dual-port RAM. In general these bits should be programmed to match the two-port RAM size on the device. It can be useful at times during development with a superset device (such as a flash superset or SE emulator device) to program the end boundary smaller than the RAM size on the superset (see Table), but to a size that matches the final production device. This sizing allows the superset to accurately emulate the A/D FIFO of the production device in terms of overruns, for example.

End Boundaries of FIFO Buffer Dual-Port RAM

BNDEND[2:0]	End Boundary
000	16 words
001	32 words
010	64 words
011	128 words
100	192 words
101	256 words
110	384 words
111	512 words

11.20 AD Buffer Control Register 3 (ADBCR3)

This register is available in compatibility and buffered modes. Figure and Table 29 describe this register.

AD Buffer Control Register 3 (ADBCR3)

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0xC4	EV DMA EN	G1 DMA EN	G2 DMA EN	EV OVR INT EN	G1 OVR INT EN	G2 OVR INT EN	EV BUF INT EN	G1 BUF INT EN	G2 BUF INT EN	Reserved			EV 8BIT	G1 8BIT	G2 8BIT	CHID
	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	U	U	U	RW-0	RW-0	RW-0	RW-0

R = Read, W = Write, S = Set, U = Undefined, *-n* = Value after reset

Table 29. AD Buffer Control Register 3 (ADBCR3) Field Descriptions

Bit	Name	Value	Description
15	EV DMAEN		Event Group DMA Enable
		0	Event group does not request DMA transfers.
		1	Event group requests DMA transfers when FIFO is RDY
14	G1 DMAEN		Group 1 DMA Enable
		0	Group 1 does not request DMA transfers.
		1	Group 1 requests DMA transfers when FIFO is RDY.
13	G2 DMAEN		Group 2 DMA Enable
		0	Event group does not request DMA transfers.
		1	Event group requests DMA transfers when FIFO is RDY.
12	EV OVR INTEN		Event Group FIFO Overrun Interrupt Enable
		0	No interrupt is generated if an event group FIFO overrun occurs.
		1	An event group interrupt is generated if the event group FIFO is overrun.
11	G1 OVR INTEN		Group 1 FIFO Overrun Interrupt Enable.
		0	No interrupt is generated if a group 1 FIFO overrun occurs.
		1	A group 1 interrupt is generated if the group 1 FIFO is overrun.

Table 29. AD Buffer Control Register 3 (ADBCR3) Field Descriptions (Continued)

Bit	Name	Value	Description
10	G2 OVR INTEN		Group 2 FIFO Overrun Interrupt Enable
		0	No interrupt is generated if a group 2 FIFO overrun occurs.
		1	A group 2 interrupt is generated if the group 2 FIFO is overrun.
8	EV BUF INTEN		Event Group FIFO Buffer Interrupt Enable
		0	No interrupt is generated if an event group FIFO threshold counter reaches 0.
		1	An event group interrupt is generated if the event group FIFO threshold counter reaches 0.
8	G1 BUF INTEN		Group 1 FIFO Buffer Interrupt Enable
		0	No interrupt is generated if a group 1 FIFO threshold counter reaches 0.
		1	A group 1 interrupt is generated if the group 1 FIFO threshold counter reaches 0.
7	G2 BUF INTEN		Group 2 FIFO Buffer Interrupt Enable
		0	No interrupt is generated if a group 2 FIFO threshold counter reaches 0.
		1	A group 2 interrupt is generated if the group 2 FIFO threshold counter reaches 0.
6–4	Reserved		Reads are indeterminate. Writes have no effect.
3	EV 8 BIT		Event Group FIFO Buffer 8-Bit Result Mode
		0	Data is read out of the event group FIFO in full 10-bit format.
		1	Data is read out of the event group FIFO in 8-bit format. That is, the 8 MSBs of the digital result (normally read from bits [9:2]) are read from bits [7:0]. Bits [9:8] read 00, and the two LSBs of the digital result are discarded.

Table 29. AD Buffer Control Register 3 (ADBCR3) Field Descriptions (Continued)

Bit	Name	Value	Description
2	G1 8 BIT		Group 1 FIFO Buffer 8 Bit Result Mode
		0	Data is read out of the group 1 FIFO in full 10-bit format.
		1	Data is read out of the group 1 FIFO in 8-bit format. That is, the 8 MSBs of the digital result (normally read from bits [9:2]) are read from bits [7:0]. Bits [9:8] read 00, and the two LSBs of the digital result are discarded.
1	G2 8 BIT		Group 2 FIFO Buffer 8-Bit Result Mode
		0	Data is read out of the group 2 FIFO in full 10-bit format.
		1	Data is read out of the group 2 FIFO in 8-bit format. That is, the 8 MSBs of the digital result (normally read from bits [9:2]) are read from bits [7:0]. Bits [9:8] read 00, and the two LSBs of the digital result are discarded.
0	CHID		Channel ID FIFO Mode
		0	Data is read out of all three FIFOs with bits [13:10] forced to 0000.
		1	Data is read out of all three FIFOs with bits [13:0] containing the ID of the A/D channel to which the digital result belongs.
			Note: In either case, the channel ID is *always* stored in the FIFO with the digital result. The CHID affects whether the channel ID is available with the data *when the FIFO is read*. Therefore, CHID can be changed dynamically.

11.21 AD Buffer Status Register (ADBUFST)

This register is available in buffered mode only. Figure 34 and Table 30 describe this register.

Figure 34. AD Buffer Status Register (ADBUFST)

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0xC8	Res					EV INT FLAG	G1 INT FLAG	G2 INT FLAG	Res	EV OVR	G1 OVR	G2 OVR	Res	EV EMP-TY	G1 EMP-TY	G2 EMP-TY
	U					RW-0	RW-0	RW-0	U	R-0	R-0	R-0	U	R-0	R-0	R-0

R = Read, W = Write, S = Set, U = Undefined, *-n* = Value after reset

Table 30. AD Buffer Control Register 3 (ADBCR3) Field Descriptions

Bit	Name	Value	Description
15–11	Reserved		Reads are indeterminate. Writes have no effect.
10	EV INT FLAG		Event Group Interrupt Flag
		0	*Read:* Event group buffer has no pending interrupt. *Write:* Writing a 0 clears the bit.
		1	*Read:* Event group buffer has pending interrupt caused by the threshold being reached. *Write:* Writing a 1 has no effect.
9	G1 INT FLAG		Group 1 Interrupt Flag
		0	*Read:* Group 1 buffer has no pending interrupt. *Write:* Writing a 0 clears the bit.
		1	*Read:* Group 1 buffer has pending interrupt caused by the threshold being reached. *Write:* Writing a 1 has no effect.
8	G2 INT FLAG		Group 2 Interrupt Flag
		0	*Read:* Event group buffer has no pending interrupt. *Write:* Writing a 0 clears the bit.
		1	*Read:* Event group buffer has pending interrupt due to threshold being reached. *Write:* Writing a 1 has no effect.
7	Reserved		Reads are indeterminate. Writes have no effect.

Table 30. AD Buffer Control Register 3 (ADBCR3) Field Descriptions (Continued)

Bit	Name	Value	Description
6	EV OVR		Event Group FIFO Overrun Flag
		0	Event group has not experienced a FIFO overrun.
		1	Event group has experienced a FIFO overrun. To clear this condition, write to ADEISR, which resets the event group FIFO.
5	G1 OVR		Group 1 FIFO Overrun
		0	Group 1 has not experienced a FIFO overrun.
		1	Group 1 has experienced a FIFO overrun. To clear this condition, write to ADISR1, which resets the Group 1 FIFO.
4	G2 OV		Group 2 FIFO Overrun
		0	Group 2 has not experienced a FIFO overrun.
		1	Group 2 has experienced a FIFO overrun. To clear this condition, write to ADISR2, which resets the group 2 FIFO.
3	Reserved		Reads are indeterminate. Writes have no effect.
2	EV1 EMPTY		Event Group FIFO Empty
		0	Event group FIFO contains valid data, and is ready to be read.
		1	Event group FIFO is EMPTY, and contains no valid data.
1	G1 EMPTY		Group 1 FIFO Empty
		0	Group 1 FIFO contains valid data, and is ready to be read.
		1	Group 1 FIFO is EMPTY, and contains no valid data.

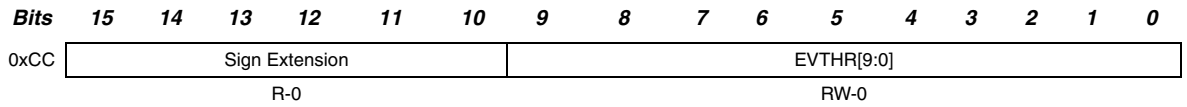
Table 30. AD Buffer Control Register 3 (ADBCR3) Field Descriptions (Continued)

Bit	Name	Value	Description
0	G2 EMPTY		Group 2 FIFO Empty
		0	Group 2 FIFO contains valid data and is ready to be read.
		1	Group 2 FIFO is EMPTY, and contains no valid data.
			Note: Follow this procedure to handle interrupts when running in continuous mode: ❑ If the interrupt is an overrun interrupt, ■ Clear the overrun interrupt flag. ■ Clear the related channel select register. ■ Return. ❑ If the interrupt is a threshold interrupt, ■ Read the expected number of values from the FIFO ■ Clear the group threshold interrupt flag ■ Return **Note that the interrupt flag is not cleared until you are finished reading the expected number of values from the FIFO.** **If there is another set of conversions written that generates a valid interrupt condition while a threshold interrupt is being serviced, the new interrupt condition will be masked out as the interrupt flag is still set. This will cause you not to read new valid data. This can be avoided by looking at the threshold counter value inside the ISR itself after you are done reading the expected number of values from the FIFO. If the counter value is negative, then there has been a new set of values written to the FIFO by the ADC, and the software can go ahead and read those values without getting another interrupt.**

11.22 AD Event Group Threshold Counter Register (ADTHREV)

This register is available in compatibility and buffered modes. Figure 35 and Table 31 describe this register.

Figure 35. AD Event Group Threshold Counter Register (ADTHREV)



R = Read, W = Write, S = Set, U = Undefined, *-n* = Value after reset

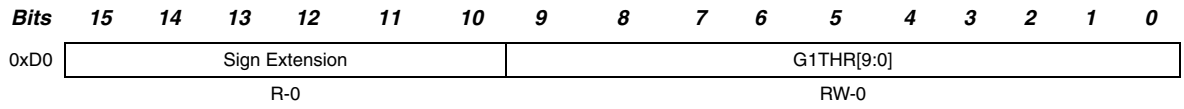
Table 31. AD Event Group Threshold Counter Register (ADTHREV) Field Descriptions

Bit	Name	Value	Description
15–10	Sign Extension		These bits always read the same as EVTHR[7].
9–0	EVTHR9:0		Event Group Interrupt Threshold Counter Before MibADC conversions begin on the event group, this register is initialized to the number of conversion results that the FIFO should contain before interrupting the CPU. When new data is written to the FIFO, this register is decremented. When data is read from the FIFO, this register is incremented. Simultaneous FIFO read and write operations leave this register unchanged. When this register reaches 0, it causes the EV BUF INT FLAG to be set, and an interrupt request is generated if the EV BUF INTEN bit is set.

11.23 AD Group 1 Threshold Counter Register (G1THREV)

This register is available in compatibility and buffered mode. Figure 36 and Table 32 describe this register.

Figure 36. AD Group 1 Threshold Counter Register (G1THREV)



R = Read, W = Write, S = Set, U = Undefined, -*n* = Value after reset

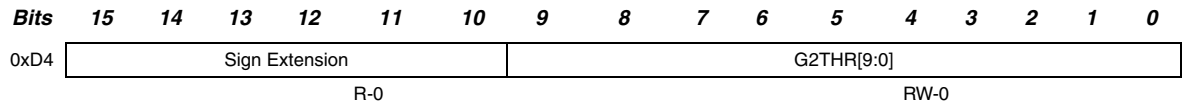
Table 32. AD Group 1 Threshold Counter Register (G1THREV) Field Descriptions

Bit	Name	Value	Description
15–10	Sign Extension		These bits always read the same as G1THR[7].
9–0	G1THR9:0		Group 1 Interrupt Threshold Counter Before MibADC conversions begin on group 1, this register is initialized to the number of conversion results that the FIFO should contain before interrupting the CPU. When new data is written to the FIFO, this register is decremented. When data is read from the FIFO, this register is incremented. Simultaneous FIFO read and write operations leave this register unchanged. When this register reaches 0, it causes the G1 BUF INT FLAG to be set, and an interrupt request is generated if the G1 BUF INTEN bit is set.

11.24 AD Group 2 Threshold Counter (G2THREV)

This register is available in compatibility and buffered mode. Figure 37 and Table 31 describe this register.

Figure 37. AD Group 2 Threshold Counter Register (G2THREV)



R = Read, W = Write, S = Set, U = Undefined, *-n* = Value after reset

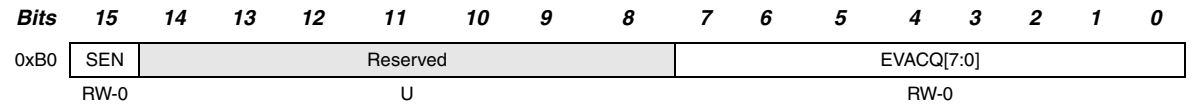
Table 33. AD Group 2 Threshold Counter Register (G2THREV) Field Descriptions

Bit	Name	Value	Description
15–10	Sign Extension		These bits always read the same as G2THR[7].
9–0	G2THR9:0		Group 2 Interrupt Threshold Counter Before MibADC conversions begin on group 2, this register is initialized to the number of conversion results that the FIFO should contain before interrupting the CPU. When new data is written to the FIFO, this register is decremented. When data is read from the FIFO, this register is incremented. Simultaneous FIFO read and write operations leave this register unchanged. When this register reaches 0, it causes the G2 BUF INT FLAG to be set, and an interrupt request is generated if the G2 BUF INTEN bit is set.

11.25 AD Sample Time Event Register (ADSAMPEV)

This register is available in compatibility and buffered mode. Figure 38 and Table 34 describe this register.

Figure 38. AD Sample Time Event Register (ADSAMPEV)



R = Read, W = Write, S = Set, U = Undefined, -*n* = Value after reset

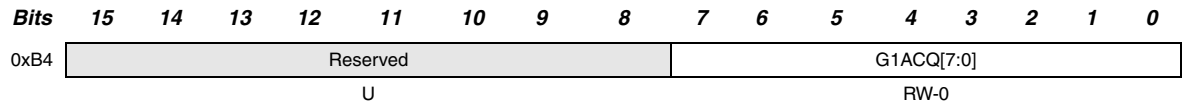
Table 34. AD Sample Time Event Register (ADSAMPEV) Field Descriptions

Bit	Name	Value	Description
15	SEN		Sample Register Enable When programmed to 1, this bit enables ADSAMPEV, ADSAMP1, and ADSAMP2 to control the sample window for their respective conversion groups and override the ACQ[1:0] bits of register ADCR1.
14–8	Reserved		Reads are indeterminate. Writes have no effect.
7–0	EVACQ7:0		Event Group Acquisition Prescale Bits These bits define the sample window (SW) for the event group. The sampling window (SW) is: SW = EVACQ + 2 The SW is specified in units of ADCLK, for fine resolution. See section 12.1, *Sample/Hold Time*, and section 12.2, *Total Conversion Time*, for an explanation of how the SW relates to the MibADC timing.

11.26 AD Sample Time Group 1 Register (ADSAMP1)

This register is available in compatibility and buffered modes. Figure 39 and Table 35 describe this register.

Figure 39. AD Sample Time Group 1 Register (ADSAMP1)



R = Read, W = Write, S = Set, U = Undefined, -*n* = Value after reset

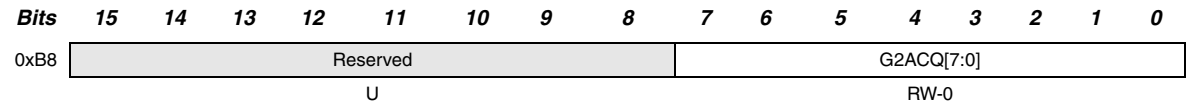
Table 35. AD Sample Time Group 1 Register (ADSAMP1) Field Descriptions

Bit	Name	Value	Description
15–8	Reserved		Reads are indeterminate. Writes have no effect.
7–0	G1ACQ7:0		Group 1 Acquisition Prescale Bits. These bits define the sample window (SW) for Group 1: SW = G1ACQ + 2 The SW is specified in units of ADCLK, for fine resolution. See section 12.1, *Sample/Hold Time*, and section 12.2, *Total Conversion Time*, for an explanation of how the SW relates to the MibADC timing.

11.27 AD Sample Time Group 2 Register (ADSAMP2)

This register is available in compatibility and buffered modes. Figure 40 and Table 36 describe this register.

Figure 40. AD Sample Time Group 2 Register (ADSAMP2)



R = Read, W = Write, S = Set, U = Undefined, -*n* = Value after reset

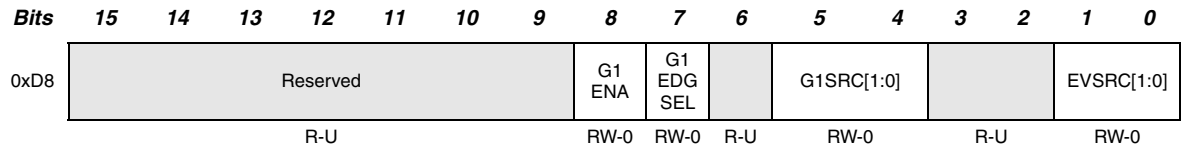
Table 36. AD Sample Time Group 2 Register (ADSAMP2) Field Descriptions

Bit	Name	Value	Description
15–8	Reserved		Reads are indeterminate. Writes have no effect.
7–0	G2ACQ7:0		Group 2 Acquisition Prescale Bits. These bits define the sample window (SW) for Group 2: SW = G2ACQ + 2 The SW is specified in units of ADCLK, for fine resolution. See section 12.1, *Sample/Hold Time*, and section 12.2, *Total Conversion Time*, for an explanation of how the SW relates to the MibADC timing.

11.28 AD Event Source Register (ADEVTSRC)

This register is available in compatibility and buffered modes. Figure 41 and Table 36 describe this register.

Figure 41. AD Event Source Register (ADEVTSRC)



R = Read, W = Write, S = Set, U = Undefined, *-n* = Value after reset

Table 37. AD Event Source Register (ADEVTSRC) Field Descriptions

Bit	Name	Value	Description
15–9	Reserved		Reads are indeterminate. Writes have no effect.
8	G1ENA		Group 1 Event Trigger Enable This bit enables Group 1 to act as an event-triggered group, instead of its default software-triggered mode.
		0	Group 1 is software triggered.
		1	Group 1 acts as a second event-triggered group. The source of the event trigger is controlled by G1SRC[1:0].
7	G1 EDG SEL		Group 1 Event Trigger Edge Select This bit determines whether the conversion Group 1, when configured for event-triggered mode, responds to the rising or the falling edge of the event input as selected by G1SRC[1:0].
		0	Trigger on high-to-low transition on source.
		1	Trigger on low-to-high transition on source.
6	Reserved		Reads are indeterminate. Writes have no effect.

Table 37. AD Event Source Register (ADEVTSRC) Field Descriptions (Continued)

Bit	Name	Value	Description
5–4	G1 SRC[1:0]		Group x Event Source Select Selects the source for Group x, if enabled:
		00	Selects the ADEVT pin.
		01	Selects internal source - TBD (that is, the timer).
		10	Selects internal source - TBD (that is, the timer).
		11	Selects internal source - TBD (that is, the timer).
3–2	Reserved		Reads are indeterminate. Writes have no effect.
1–0	EVSRC[1:0]		Event Group Source Select Selects the event source for the A/D event group:
		00	Selects the ADEVT pin.
		01	Selects internal source - TBD (that is, the timer).
		10	Selects internal source - TBD (that is, the timer).
		11	Selects internal source - TBD (that is, the timer).
			Note: For the AD event group, the polarity of the input signal that triggers the conversion is determined by ADCR2.8 EV EDG SEL. This affects the polarity of the trigger event from the ADEVT pin and the other three sources selected by EVxSRC[1:0].
			Note: Assume that the three timer options for the event group and the three timer options for group 1 are the same. That is, there are four event inputs to the AD module (ADEVT + three timer options) *not seven inputs (not ADEVT + six timer options).*

12 MibADC Timing

The programmer of the MibADC must understand two timing parameters:

- ❏ Sample/hold time
- ❏ Total conversion time

Section 12.1 describes the sample/hold time, which occurs during the first portion of a conversion. During this time, the MibADC acquires the signal on the selected analog input pin.

Section 12.2 describes the total conversion time, which is the time required for a complete sample/hold/convert to occur.

12.1 Sample/Hold Time

The MibADC must be programmed so that the sample/hold time meets two requirements:

- ❏ Minimum start hold time - $\mathbf{T_{(STHMIN)}}$
- ❏ Minimum acquisition time - $\mathbf{t_{ACQMIN}}$

The minimum start hold time is an intrinsic parameter of the analog/digital core, and is required to allow the internal analog circuitry to reset between two successive conversions. It sets a lower bound on the sample/hold time. A typical value for this parameter is 1μs; but refer to the specific device datasheet for actual timing requirement $\mathbf{T_{(STHMIN)}}$.

The minimum acquisition time is a function of the external source impedance driving the analog input pin being converted. It also depends upon the internal switch resistance and capacitances of the MibADC, but these parameters are fixed.

Section 12.2 describes how to determine the minimum acquisition time ($\mathbf{t_{ACQMIN}}$) using (EQ 9).

The worst case sample/hold time for the MibADC is a function of three variables, the ICLK period $t_{c(ICLK)}$, the ADCLK prescaler (P_s) and the sample window (SW).

The ADCLK prescaler (P_s) is determined by PS[2:0]; see section 11.1, *AD Control Register 1 (ADCR1)*.

The sample window (SW) is controlled by one of these bitfields:

- ACQ[1:0], see section 11.1, *AD Control Register 1 (ADCR1)*.
- EVACQ[7:0], see section 11.27, *AD Sample Time Group 2 Register (ADSAMP2)*
- G1ACQ[7:0], see section 11.26, *AD Sample Time Group 1 Register (ADSAMP1)*
- G2ACQ[7:0], see section 11.27, *AD Sample Time Group 2 Register (ADSAMP2)*

Once P_s, $t_{c(ICLK)}$, and SW are known, the worst-case MibADC sample/hold time is determined either by (EQ 2) or by (EQ 3), depending upon the value of P_s:

$$T_{(WCSH)} = \left(SW - \frac{1}{2}\right) \times t_{c(ICLK)} \qquad P_s = 1 \qquad \text{(EQ 2)}$$

$$T_{(WCSH)} = ((SW \times P_s) - (P_s - 1)) \times t_{c(ICLK)} \qquad (P_s \in \{2, 3, \ldots 8\}) \qquad \textit{(EQ 3)}$$

The MibADC must be programmed (by selecting P_s,SW values) such that $T_{(WCSH)}$ is longer than both $\mathbf{T_{(STHMIN)}}$ and $\mathbf{t_{ACQMIN}}$.

12.2 Total Conversion Time

The total conversion time, defined as the time between the start of one conversion to the start of the next conversion, depends upon several factors:

- The ADCLK prescaler (P_s)
- The SW
- The initial seek delay (ISEEK)

The ADCLK prescaler (P_s) is determined by PS[2:0], see section 11.1, *AD Control Register 1 (ADCR1)*.

The sample window (SW) is controlled by one of these bitfields:

- ACQ[1:0], see section 11.1, *AD Control Register 1 (ADCR1)*.
- EVACQ[7:0], see section 11.27, *AD Sample Time Group 2 Register (ADSAMP2)*
- G1ACQ[7:0], see section 11.26, *AD Sample Time Group 1 Register (ADSAMP1)*

- G2ACQ[7:0], see section 11.27, *AD Sample Time Group 2 Register (ADSAMP2)*

The parameter ISEEK is the initial seek delay incurred to find the first channel in the group, and should be set to the number of that channel. Thus, if the first channel in a group is channel 0, set ISEEK to 0. But if the first channel is 5, then ISEEK is 5.

The signal to start a series of conversions is triggered off of the peripheral's ICLK (i.e., an event or software group trigger) and may occur during any phase of ADCLK when P_s (ratio of ADCLK/ICLK period) is 2,3,...8. Without knowing the phase relationship between ADCLK and the ICLK cycle on which the start signal is asserted, it is only possible to bound the first conversion in a series by (EQ 4) (minimum) and (EQ 5) (maximum):

$$t_{(SHCMIN)} = \left((SW + 10) \times P_s + \left(P_s - \text{int}(\frac{P_s}{2})\right) + 3 + ISEEK\right) \times t_{c(ICLK)} \quad \textit{(EQ 4)}$$

$$t_{(SHCMAX)} = \left((SW + 11) \times P_s + \left(P_s - \text{int}(\frac{P_s}{2})\right) + 2 + ISEEK\right) \times t_{c(ICLK)} \quad \textit{(EQ 5)}$$

However, subsequent conversions in the same group will be synchronized to ADCLK based upon the end of the previous conversion. They will have a period that is described either by (EQ 5), (EQ 6), or (EQ 7):

$$t_{(SHC)} = t_{(SHCMIN)}, \quad \text{when } P_s = 1, \text{ otherwise} \quad \textit{(EQ 6)}$$

$$t_{(SHC)} = ((SW + 12) \times P_s) \times t_{c(ICLK)}, \quad \text{when } \left(\text{int}\left(\frac{P_s}{2}\right) \leq (SEEK + 3)\right) \quad \textit{(EQ 7)}$$

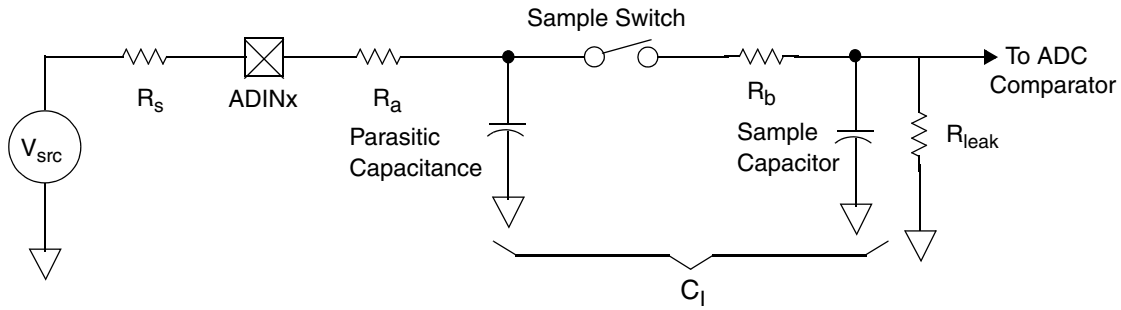
$$t_{(SHC)} = ((SW + 11) \times P_s) \times t_{c(ICLK)}, \quad \text{when } \left(\text{int}\left(\frac{P_s}{2}\right) > (SEEK + 3)\right) \quad \textit{(EQ 8)}$$

13 Analog Input Modeling

To determine how fast a conversion can be performed, it is necessary to model the MibADC input as shown in Figure 42. The *selected* MibADC input channel can be modeled as a 250-ohm resistor (R_I in Figure 42) in series with a 10-pF capacitor to ground during the conversion period. The capacitance increases to 30 pF (C_I in Figure 42) at all other times. Figure 42 shows the basic model of the driving source and an MibADC input channel. The driving source consists of the voltage source V_{src} and source resistance R_s. Inside the MibADC, R_a represents the resistance of the *input switch*. R_b represents the resistance of the *sample switch* during *on* time. C_I is the total capacitance seen by the driving source when the switch is closed. Let $R_I = R_a + R_b$.

A total resistance of $R_s + R_I$ is seen by the source, V_{src}. When the sample multiplexer selects the channel being modeled, the *sample switch* is closed and the total capacitance, C_I, must be charged via $R_s + R_I$. The respective R_IC_I time constant establishes the minimum time required to sample the input voltage, V_{src}.

Figure 42. Input Source and MibADC Model



R_{leak} is the leakage resistance of the sample capacitor and switch during *hold* time. R_{leak} limits the maximum time that can be spent during conversion without losing accuracy.

The MibADC requires a finite amount of time for the voltage placed on the sample capacitor to become stable. If the full accuracy is expected from the MibADC, then the voltage must settle to within 1/2 LSB. For the 10-bit MibADC, the required time is about 8 RC time constants and can be calculated as follows:

$$\frac{1}{2^{(10+1)}} = e^{\frac{-t_{ACQMIN}}{(R_S + R_I) \cdot C_I}} \qquad \textit{(EQ 9)}$$

Where:

R_s = Source resistance of external circuit driving the MibADC channel, assumed to be 0 here

R_I = MibADC channel input resistance, 250 ohms typical

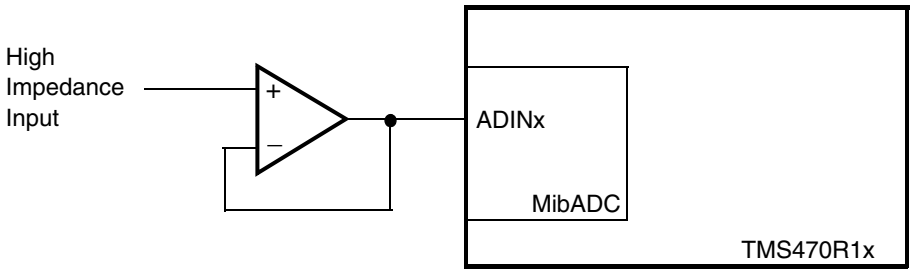
C_I = MibADC channel input capacitance during sampling time, 30 pF

With R_s zero and the typical switch characteristics above, this yields (EQ 10):

$$t_{ACQMIN} \geq 57\text{ns} \qquad \textit{(EQ 10)}$$

Assuming that the source resistance R_s is 0 usually implies that the source is an active circuit such as an op amp as in Figure 43, or it can mean that there is a large capacitance at the input pin that has been charged to within 1/2 LSB of the correct voltage. The potential advantage of using this capacitor is that it is being continuously charged by the source, giving it at least 16 times longer to charge than the MibADC sample capacitor.

Figure 43. Driving MibADC Input Channel with Low Impedance



When you connect an op amp in a unity gain configuration as in Figure 43, one that is unity-gain compensated must be used to avoid oscillation. One such op amp is the TI TLE2227A.