Safe readout of SDADC converted data on SPC58 devices

Introduction

SPC58 devices integrate the sigma-delta analog-to-digital converter (SDADC). This peripheral requires careful handling when reading converted data to prevent faults.

This document describes the recommended procedure for reading data converted by the SDADC. It applies to all SPC58 devices that include the SDADC. The intended readers are users and software developers working with the SDADC.

1 How to access converted data

Users can access SDADC converted data through either the converted data register (CDR) or the latest data register (LDR). The choice depends on the intended use of the SDADC.

The SDADC includes a FIFO that stores converted data. With the FIFO enabled, the SDADC can run continuously while software or DMA reads data asynchronously, without losing samples. Users can enable or disable the FIFO by setting the FE field in the SDADC_FCR register.

When the FIFO is enabled, the SDADC buffers converted samples before exposing them to software. In this mode, the CDR provides deterministic and safe access to the acquired data.

When the FIFO is disabled, direct access to the CDR becomes timing sensitive. If a read occurs while the decimation filter updates the CDR, different read paths may return values that are not coherent. This can lead to inconsistent data and, in some cases, a fault condition.

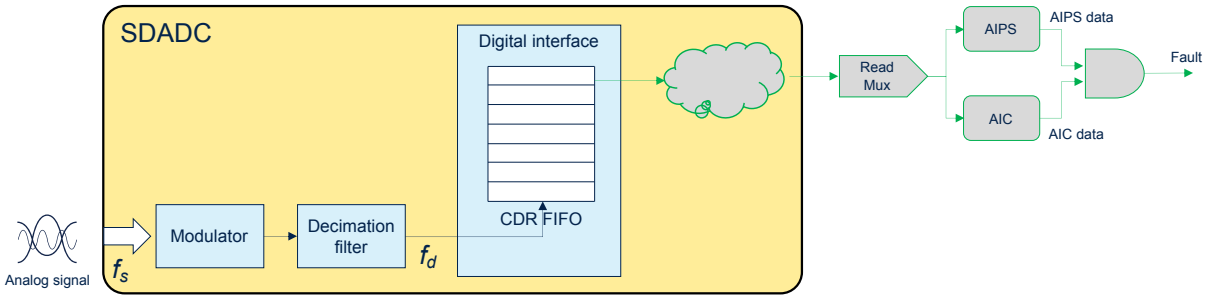
Next section describes the recommended readout strategies and explains why the LDR must be used when the FIFO is disabled.

2 FIFO enabled

When the FIFO is enabled, converted data is stored in the FIFO, and then read through the CDR. The application must wait until the FIFO is not empty (see data FIFO empty flag) before reading a converted sample. This ensures that the access occurs only when valid data is available.

In this mode, the CDR can be used for continuous acquisition, and DMA support is also possible if the system is properly configured.

Figure 1. CDR read path with FIFO enabled



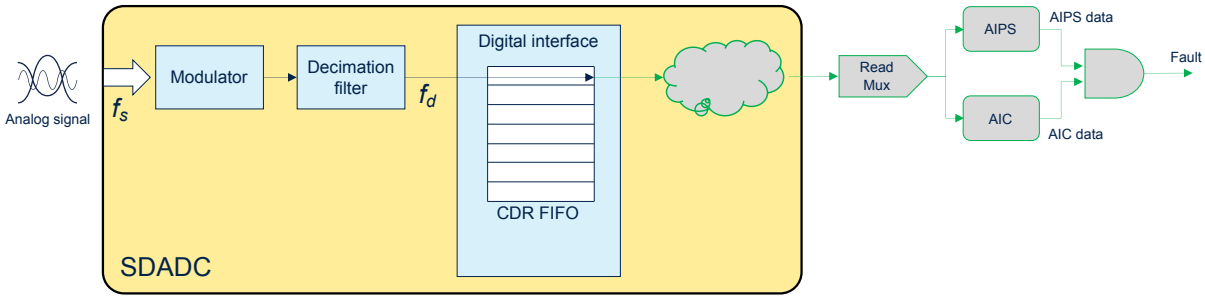
3 FIFO disabled

When the FIFO is disabled, the converted data is no longer protected by the same buffering mechanism.

A read of the CDR may overlap with a data update generated by the decimation filter. This timing circumstance may cause different read paths to sample different values. As a result, the AIPS and AIC modules may not capture the same data, which can trigger a fault (refer to Figure 2).

Therefore, the CDR must not be used for readout when the FIFO is disabled.

Figure 2. CDR read path with FIFO disabled



3.1 Why CDR reading may fail

The issue arises from the interaction between two design features.

First, the SDADC operates with two clock domains: SDADC_CLK and AIPS_CLK. Second, when a master reads ADC or other peripheral registers, the access passes through AIPS and its replica, AIC. If the values observed by AIPS and AIC do not match, the system flags a fault.

When the FIFO is enabled, the decimation filter output, which is driven by SDADC_CLK, passes through the FIFO before reaching the CDR register (refer to Figure 1). In this configuration, the data remains stable with respect to AIPS_CLK when a master reads through CDR, and no problem occurs.

When the FIFO is disabled, the decimation filter output goes directly to the CDR register (refer to Figure 2). In this case, depending on the exact timing of the data transfer into CDR and the subsequent read, the data may not be stable with respect to AIPS_CLK. As a result, AIPS and its replica may capture different values, and the system detects this condition as a fault.

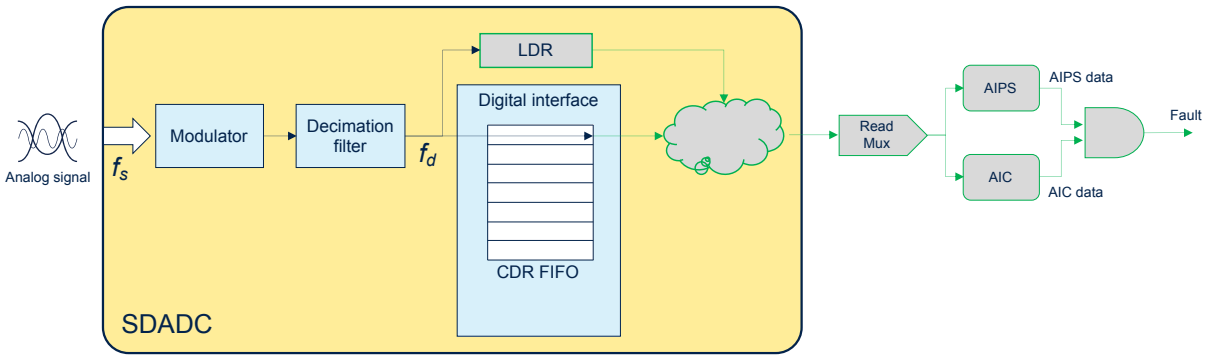
By contrast, the data path through LDR is properly synchronized. Therefore, reading LDR does not trigger this type of fault.

3.2 Recommended solution: read the LDR

To prevent faulty behavior associated with CDR reads when the FIFO is disabled, the application must read converted data through the LDR. This field always contains the most recent converted sample and provides a safe, coherent readout point.

Unlike the CDR, the LDR is intended for point-in-time access and avoids the issues that can affect direct CDR reads in FIFO-disabled mode.

Figure 3. **LDR read path with FIFO disabled**

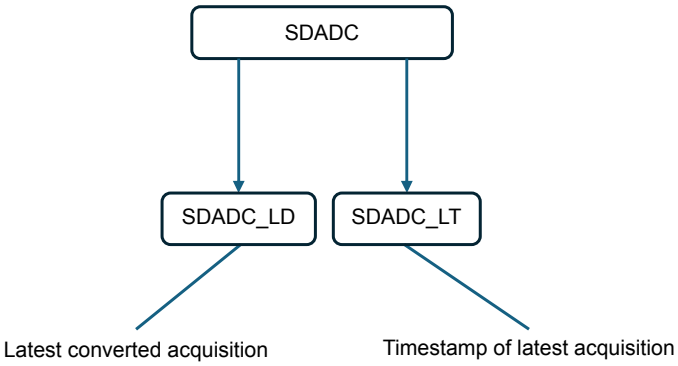


3.3 Acquisition of timestamp information

When the application reads the converted value from the LDR, it can also retrieve the acquisition time by reading the latest timestamp register (SDADC_LTREG). This register stores the timestamp associated with the latest acquisition and can be used to correlate the data value with its sampling time.

This is useful when the application needs both the most recent converted sample and the corresponding time information.

Figure 4. **LDR latest converted sample and SDADC_LTREG timestamp acquisition**



4 Reading strategy summary and conversion steps

The recommended choice of the readout method depends on the application requirements:

- If the application requires all converted samples, the FIFO should be enabled, and the data should be read from the CDR when the FIFO is not empty
- If the application requires the converted value at a specific time (for example, every ms) and loosing samples is acceptable, the FIFO should be disabled, and the data should be read from the LDR

This rule separates continuous acquisition from point-in-time acquisition and avoids the faulty behavior observed with direct CDR reads in FIFO-disabled mode.

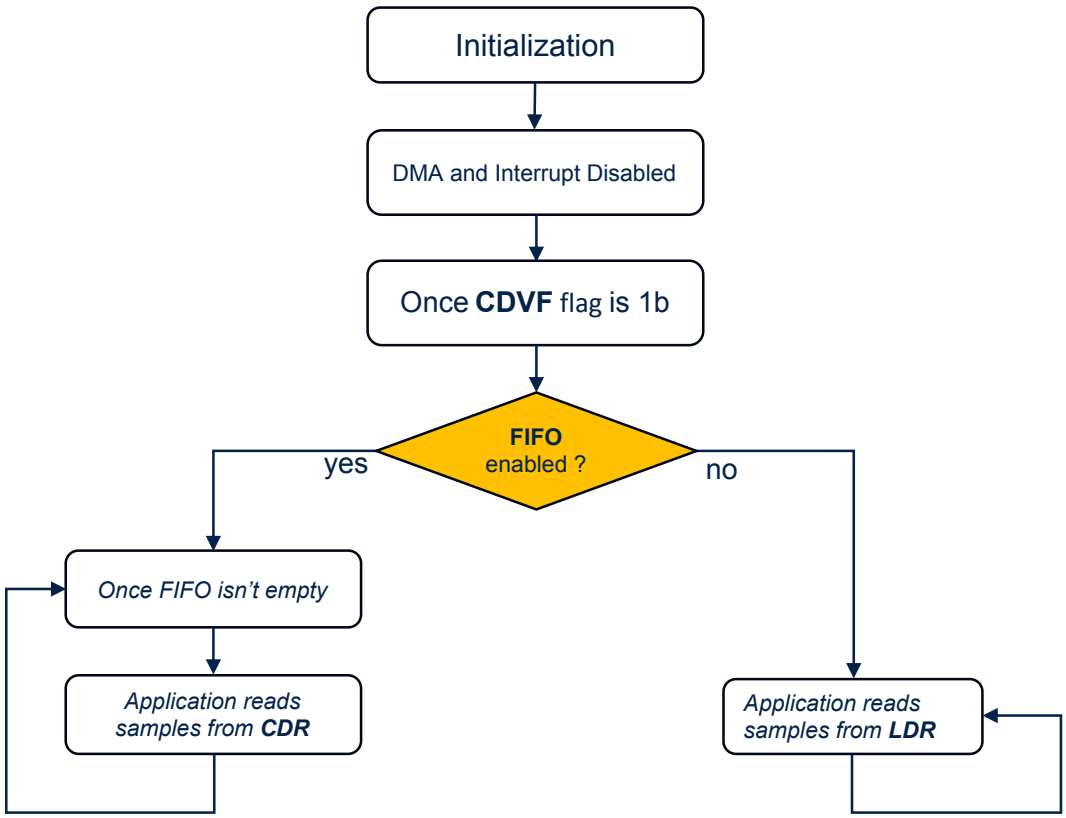
This text below details the procedure outlined in the “Data conversion step” section of the reference manual for reading converted data correctly.

After software generates a reset event by writing 0x5AF0 to the RESET_KEY field of the RKR register, software must wait for the CDVF flag to assert. This flag indicates that the SDADC output data is valid and remains asserted until software triggers the next reset event.

At this point, if the FIFO is disabled, software can begin reading the acquired data from LDR. If the FIFO is enabled, software must first wait until the FIFO is not empty and then read CDR.

Next section lists the recommended steps for a safety acquisition.

Figure 5. LDR and SDADC_LTREG relationship



4.1 Data conversion steps

To acquire data from the SDADC, when the FIFO is enabled follow this sequence:

1. Enable the SDADC by setting MCR[EN]
2. Configure the MCR register to select the required mode, polarity, common-mode voltage, input gain, and decimation rate
3. Enable the FIFO
4. Enable the high-pass filter, if required
5. Select the analog channel to be converted. For AC-coupled applications, configure the bias for each channel as needed
6. Set the OSD delay according to the required SDADC startup time or the latency after reset exit
7. Generate an SDADC reset event by writing 0x5AF0 to the RESET_KEY field in the RKR register
8. Wait for the assertion of the CDVF flag
9. Wait until the FIFO is not empty (the DFEF flag is 0b)
10. Read the converted data from the CDR register using the CDATA field
11. For each new acquisition, go to step 9

To acquire data from the SDADC, when the FIFO is disabled follow this sequence:

1. Enable the SDADC by setting MCR[EN]
2. Configure the MCR register to select the required mode, polarity, common-mode voltage, input gain, and decimation rate
3. Disable the FIFO
4. Enable the high-pass filter, if required
5. Select the analog channel to be converted. For AC-coupled applications, configure the bias for each channel as needed
6. Set the OSD delay according to the required SDADC startup time or the latency after reset exit
7. Generate an SDADC reset event by writing 0x5AF0 to the RESET_KEY field in the RKR register
8. Wait for the assertion of the CDVF flag of the SDADC_SFR register
9. Read the converted data from the SDADC_LDREG register using the LDR field
10. For each new acquisition, go to step 9

5 Conclusion

This document specifies how to safely read data acquired by the SDADC integrated into SPC58 devices.

When the FIFO is enabled, the CDR can be read safely once the FIFO is not empty.

When the FIFO is disabled, the CDR must not be used because a read may overlap with a data update and return inconsistent data or cause a fault. In this case, the application must read the LDR, which provides a safe and coherent view of the latest converted sample.

Appendix A Acronyms and reference documents

Acronyms

AIC	Peripheral bridge (AIPS) integrity checker
AIPS	Peripheral bridge
CDR	Converted data register
CDVF	Converted data valid flag
DMA	Direct memory access
FCR	FIFO control register
FIFO	First input first output
LDR	Latest data register
LTREG	Latest timestamp register
MCR	Module configuration register
RKR	Reset key register
SDADC	Sigma-delta analog-to-digital converter

Reference documents

[1] Reference manual *SPC58xNx 32-bit Power Architecture microcontroller for automotive ASILD applications* (RM0421)

[2] Datasheet *32-bit Power Architecture microcontroller for automotive ASIL-D applications* (DS11734)

Revision history

Table 1. **Document revision history**

Date	Revision	Changes
07-May-2026	1	Initial release.

Contents

List of tables

List of figures