Understanding and Extending Safety Operation in a Sigma-Delta ADC

By **Miguel Usach Merino**

Share on

Abstract

New international standards and regulations have accelerated the need for safety systems in industrial equipment. The objective of functional safety is to protect people and assets from harm. This is achieved through the use of safety functions that target specific hazards. Safety functions consist of a chain of subsystems including sensor, logic, and output blocks and so require system-level and integrated circuit level expertise to deliver an IC with the right set of features. This articles explores the AD7770 Σ-Δ ADC as an example of a high performance IC conceived and designed to provide an advanced set of features in both the analog and digital domains that simplifies the design of safety systems.

Introduction

Paraphrasing one of the Murphy's laws: "If the possibility exists of several things going wrong, the one that will go wrong is the one that will do the most damage."

A system that could produce a direct threat to human life, or an indirect threat, like a failure in machinery, must be designed to minimize the probability of failure and its consequential negative effects. To guarantee that the level of probabilistic random and deterministic failure is kept as low as possible, a specific design methodology must be followed. In the industry, this design methodology is called functional safety. This methodology requires a meticulous analysis of the system to identify any potentially hazardous situations and the application of best practices to bring the risk of malfunction down to tolerable levels in the component, subsystem, or system, such as unsafe states (that is, the voltage is too high, or diagnostic failure).

The idea behind functional safety is to keep the system in a safe state when an error is detected, like disconnecting the active outputs if the conversion results from an external sensor are out of bonds.

IEC-61508 is the standard reference for functional safety design in industrial equipment, and has been adapted/interpreted for different industries, like the ISO-26262 for automotive, or IEC-61131-6 for programmable controllers.

Designing to a functional safety standard can be quite tedious as a top-down meticulous analysis must be done, from the overall system description to the internal functional blocks of the components used. This analysis is necessary to guarantee enough level of protection to avoid any hazardous situation and to minimize the probability of the occurrence of undetected errors. A functional safety system should be designed in such a way that the system is capable of detecting any error and reacting fast enough to minimize the probability of the hazardous situation, as shown in Figure 1.

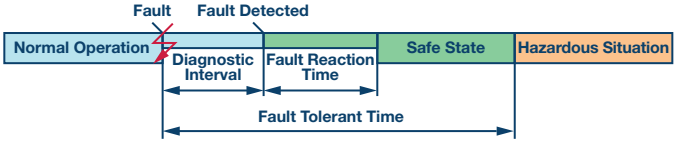

Figure 1. Reaction time in a functional safety system.

How to Design a Functional Safety System

The first step is a hazard analysis to identify the ways that somebody could get hurt. After the analysis of those situations, the system should be designed in such a way that hazardous situations can be avoided. If there is an unavoidable situation, add a functional system to detect the unsafe state and bring the system to a safe situation.

To illustrate the problem, let's assume the hypothetical system as shown in Figure 2. Depending on the tank temperature, a valve connected to a tank is opened a percentage to minimize the risk of explosion. A DAC controls the aperture of the valve through a motor. The system described is open-loop.

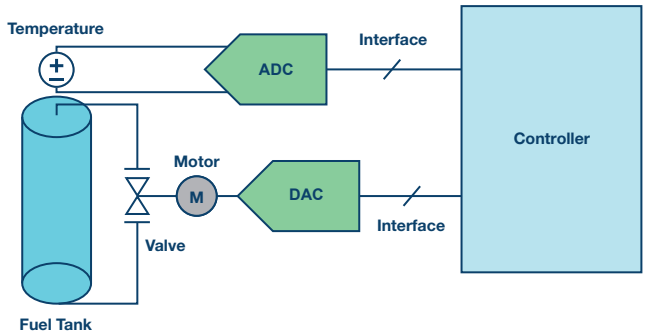

Figure 2. Open-loop valve control system signal chain.

The hazard analysis reveals two situations that could produce an unsure state:

- The temperature is incorrectly measured. Consequently, the aperture of the valve is incorrect.
- The DAC fails to open/close the valve correctly.

The next step is to evaluate the risk associated with each hazard as,

$$Risk = probability\ of\ occurrence\ of\ harm \times severity\ of\ the\ harm$$

Once the risk is identified, the next step is to design a functional safety system capable of reducing the risk to a tolerable level.

IEC-61508 defines four safety integrity levels (SIL) that define the level of risk reduction achieved by a safety function. There are two different target probabilities: the failure on demand, which applies to systems that are in stand-by until an event triggers (airbags are a good example), and probability of failure per hour, which applies to systems that are constantly operating, as could be the case in the previous example. Table 1 summarizes rough equivalences between SIL according to IEC61508, ISO 26262 (ASIL, automotive), and the avionics standard for expected failures on demand and per hour.

Table 1. Risk Levels Approximations for Different Standards

Probability of Failure on Demand	Probability of Failure Per Hour	Standard		
		IEC 61508 SIL Level	Automotive	Avionics
0.1 to 0.01	10^{-5} to 10^{-6}	1	A	D
0.01 to 0.001	10^{-6} to 10^{-7}	2	B	C
0.001 to 0. 0001	10^{-7} to 10^{-8}	3	C/D	B
0.0001 to 0.00001	10^{-8} to 10^{-9}	4		A

SILs are based on the required reduction and minimization of an undetected failure generating a malfunction on the system and potentially triggering an undesirable situation.

What Are Diagnostic Coverage Requirements?

The probability of undetected failure decreases with the increment of the diagnostic coverage. If the system can provide 99% diagnostic coverage, SIL3 can be achieved; for 90% diagnostic coverage, SIL2 can be claimed. If the coverage is only 60%, SIL1 can be achieved. In other words, the occurrence of undetected errors decreases with the level of redundancy.

The easier way to achieve SIL2 or SIL3 is by employing components already qualified for this grade of protection. This is not always possible as these types of components target specific applications, which may not be identical to your circuit or system. Consequently, the assumptions applied to qualify the device may not apply, and the level of protection may not be the same.

Another approach for achieving high diagnostic coverage is by applying redundancy at the component level. In this case the error detection is not done directly, but indirectly by comparing two (or more) outputs that should be the same. However, this approach will increase the power consumption, the area, and, probably more importantly, the final cost of the system.

Increasing Error Detection and Redundancy at the Component Level

A common source of error is the data transmission in the external interface; if any single bit is corrupted during the transmission, the data can be misinterpreted by the receiver and can generate an undesirable situation. To calculate the total error that occurs in transmitting data, the BER (bit error rate) can be used. The BER indicates the number of bits corrupted due to noise, interferences (EMC), or any other physical reason.

$$BER = \frac{bits\ corrupted}{bits\ transmitted}$$

The BER can be physically measured in the system. Generally, this number is defined in many standards, as is the case in HDMI®, or an estimated value can be used. The minimum standard BER for modern data traffic is 10^{-7}. This number may be considered too pessimistic for many applications, but it can be used for reference purpose.

A BER of 10^{-7} means that 1 bit in every 10 million bits will be corrupted. For a SIL3 system, the target maximum probability of errors per hour is 10^{-7}. If our system transmits 32 bits of data between the ADC to the controller with an output data rate of 1 kSPS, then in one hour it will transmit:

$$bits\ per\ hour = 32 \times 1000 \times 3600 = 115{,}200{,}000\ bits$$

In this case, the error rate will increase up to $1.5e^{-5}$, and this is only the contribution from one interface; the total contribution of transmission errors should be kept to between 0.1% to 1% of the total error budget.

In this case, the error can be detected by adding a CRC algorithm. The number of bits corrupted that can be detected is defined by the Hamming distance of the CRC polynomial, such as $X^8 + X^2 + X + 1$, which has a Hamming distance of 4 and is capable of detecting up to three corrupted bits per frame transmitted. Table 2 summarizes the probability of error based on the number of bits transferred per hour for a CRC Hamming distance of 4 at different bits per hour, when transferring 32 bits of data plus 8 bits CRC.

Table 2. Probability of Error for a CRC Hamming Distance of 4

Number of Data Bits Per Hour	Probability of Undetected Error Per Hour
144,000,000	$2e^{-14}$
432,000,000	$6e^{-14}$
2,160,000,000	$3e^{-13}$

The level of diagnostic using the CRC can be augmented by reading back the register that was written, and confirming that the data has been correctly transferred. This action will increase the level of diagnostic, but the level of error detection on the CRC polynomial used must be capable of detecting the expected number of bits corrupted based on the BER probability.

What Can Be Done to Minimize the Failure Probability?

A manufacturer who claims that a component has been designed for a functional safety system should be able to provide the FIT and, more importantly, failure modes, effects, and diagnostic analysis (FME(D)A). This data is used to analyze the IC in a specific application to calculate diagnostic coverage (DC), safe failure fraction (SFF), and dangerous failures rates for the system.

The FIT is a measurement of the reliability of a device. FIT for an IC can be calculated based on accelerated life testing or based on industry standards such as IEC62380 and SN29500 where the average operating temperature in the application, package type, and number of transistors are considered to generate a FIT prediction. The FIT does not provide any information about the root cause of the failure, just a reliability prediction for the device. Generally speaking, unless each functional block can be checked, directly or indirectly, the final error probability will be too high to meet the SIL targets for any SIL2 or SIL3 safety functions.

The objective of the FME(D)A is to provide a comprehensive document covering the analysis of all the blocks implemented in the silicon, the consequences of a failure in the block directly or indirectly, and the different mechanism or methods that allow detection of the failure. As previously mentioned, those analyses are done based on a given signal chain/application, but the level of detail provided should be high enough to easily generate an FME(D)A analysis for a different system/application.

What Can Go Wrong in a Σ-Δ ADC?

A general analysis of a Σ-Δ ADC highlights multiple sources of errors due to the internal complexity of this device, such as:

- Reference disconnected/damage
- Input/output buffers/PGAs damage
- ADC core damage/saturation
- Incorrect internal regulator supply
- Incorrect external supply

These are only some problems that could generate a failure in a device block, but there are other sources of failure that may not be as obvious as the previously listed ones, such as:

- Internal bonding damage
- Bonding short-circuit with adjacent pin
- Leakage current increment

For example, could the component detect if the V_{REF} leakage current increases, generating a drop on the internal reference voltage?
To check this type of malfunction, the ADC should be capable of selecting between different references for conversion and have the V_{REF} as an input for conversions.

How could you detect if the internal fuses have regrown or are otherwise corrupted, which could load an incorrect configuration at power-up? Those are examples of things that can go wrong even if the probability is really low. All of the potential failures, especially the ones that can be very rare, and the way that they can be detected (if any), must be well documented in the FME(D)A document. This document summarizes the failures and the assumptions made based on a specific application and/or configuration to maximize the detection and minimization of undetected errors.

ADI's modern ADI Σ-Δ ADCs, like the AD7770, AD7768, or AD7764, implement multiple diagnostic detectors to increase the fault tolerance protection, and to detect functional errors in both digital and analog blocks. Example of these blocks are:

- CRC checker for the fuses, registers, and interfaces
- Overvoltage/undervoltage detectors
- Reference and LDO voltage detectors
- Internal fixed voltage for PGA gain testing
- External clock detector
- Multiple reference voltage sources

In addition to these features, the AD7770 ADC integrates an auxiliary 12-bit SAR ADC that can be used to increase the diagnostic capability of the device, with uses such as:

- Implementing an alternate architecture that can offer some benefits like providing a different level of immunity to EMC
- It is powered through different supply pins, which can be used as a reference
- It is fast enough to monitor the eight Σ-Δ channels for a single conversion of a Σ-Δ channel as a monitor but with a different accuracy
- It provides conversion results using a different serial interface (SPI)
- Provides access to all internal voltage nodes for diagnostics like the external supplies, V_{REF}, V_{CM}, LDO output voltage, or internal voltage reference

Figure 3 shows the internal block diagram of the AD7770 ADC. The blocks that include an internal monitor are highlighted in purple, the blocks highlighted in green can be actively monitored, and the blocks highlighted in blue contain both internal and active monitoring functionality.

Conclusion

Functional safety consists of reducing the mathematical probability of undetected errors by increasing system/block monitoring and diagnostic coverage. The easier way to increment the coverage is by adding redundancy, but this penalizes the system in multiple ways, especially cost. Recent ADI Σ-Δ ADCs, like the AD7124 or AD7768, implement many internal error detectors, which simplify the design of a functional safety system, keeping the overall complexity low compared with other solutions. The AD7770 is a good example of a precision Σ-Δ ADC design that is ahead what's possible due to its integrated monitoring and diagnostic capabilities, which include an internal redundant converter to maximize diagnostic coverage.

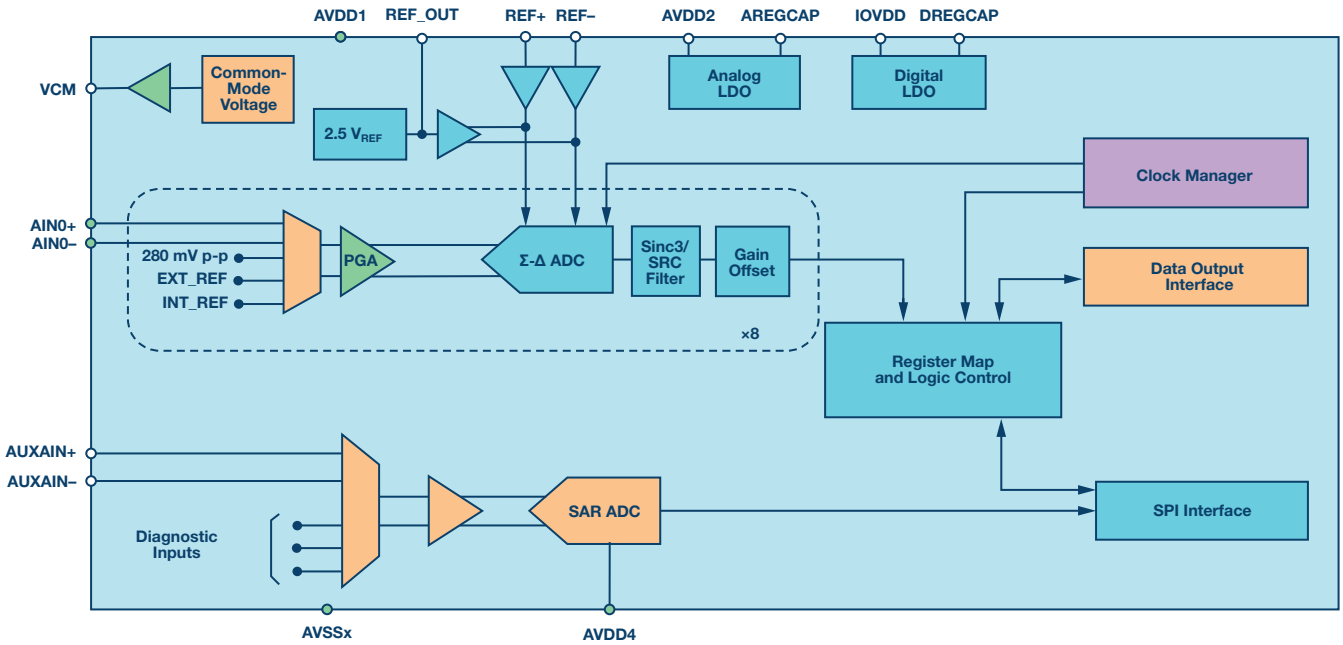

Figure 3. AD7770 ADC's diagnostic and monitoring blocks.

Miguel Usach Merino [miguel.usach@analog.com] received his degree in electronic engineering from Universitat de Valencia. Miguel joined ADI in 2008 and works as an applications engineer in the Linear and Precision Technology Group in Valencia, Spain.

Miguel Usach Merino

Also by this Author:

Integrated Capacitive PGAs in ADCs: Redefining Performance

Volume 50, Number 3