*Application Note*

# *ADC128S102QML-SP Application Advisory*

## ABSTRACT

The ADC128S102QML-SP may display sparkle codes at the output of the device at an infrequent rate. A sparkle code is an erroneous output code that has a predictable output value and will only appear under specific input conditions. A more complete definition and description of a sparkle code is provided in this document. This application report provides background on the sparkle code behavior and provides empirical data across various device configurations and temperatures. From the data, it was found that a sparkle code occurrence is in the order of parts-per-billion over process and temperature. Also, the ADC128S102QML-SP design differs from the catalog version of the device and these design differences make the ADC128S102QML-SP susceptible to sparkle codes, whereas the catalog version of the device is not susceptible to sparkle codes.

## Table of Contents

## List of Figures

## List of Tables

## Trademarks

All trademarks are the property of their respective owners.

# 1 Introduction

The ADC128S102QML-SP may exhibit code errors called sparkle codes at the output of the device. A sparkle code is an erroneous output that occurs very infrequently under very specific input conditions. A code error is defined as an error in the output code of an analog to digital converter (ADC) that exceeds a defined error threshold. The acceptable range of an ADCs output is commonly defined to include performance parameters including, but not limited to, offset, gain, and noise. If the output exceeds the expected amplitude of the ADC's acceptable error threshold, this is considered an error code. A sparkle code is a single erroneous output code either above or below the threshold that can be easily identified in the presence of noise, in the case of the ADC128S102-SP, by a minimum delta of 0xFF from the expected output code. Another way to explain this definition is that the observed error amplitude occurs with a probability exceeding the expected probability of the error amplitude given the ADC's assumed Gaussian distributed noise. That is, sparkle codes are clearly visible and easily recognizable in data sample sets and data sample plots as shown in Figure 1-1.

An important characteristic that defines sparkle code is that it will only occur when the expected converter output code is at very specific values. The erroneous sparkle code measured will occur at predicted values, available in Table 3-1. For example, if the expected output code is in the range of 0x0FF – 0x100 than a sparkle could occur where the output value is 0x1FF. A sparkle code can only occur under certain output value conditions and has a predictable sparkle code value. The physical mechanism causing the sparkle code will be described in the Root Cause Analysis section.

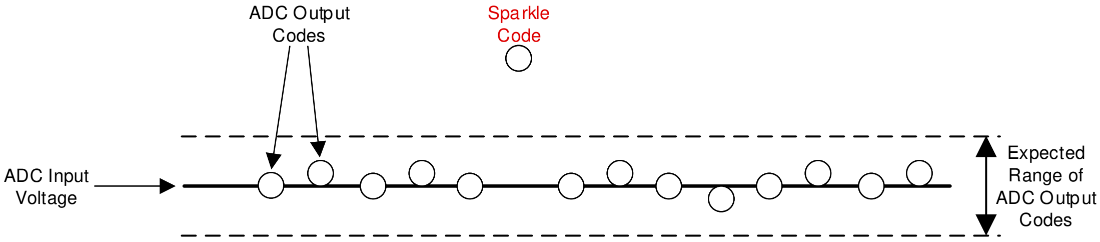

**Figure 1-1. Example of Sparkle Code Outside of Expected Output Range**

## 2 Comprehensive Summary

To understand the occurrence of the sparkle code, empirical data was collected across multiple devices and multiple configurations, using a test solution that maintains the precise input conditions required to elicit sparkle codes. A detailed evaluation of the results is available further in this document, and a high-level overview of the results is provided in this section.

The custom test solution was created for this project, the hardware was defined to be able to drive the input of the ADC, and use a socket to easily change devices. The most sophisticated aspect of the test solution is the closed servo loop controlling the input voltage to the device. This provided the precise input needed to maintain the input at the necessary voltage for the converter output code to be within half an LSB of the required code transition for multiple hours to allow monitoring for sparkle codes. Details on the test solution used is found later in this document.

A frequency of sparkle code occurrence has been observed to be in the order of parts per billion (ppb) across all configurations and temperatures. At ambient room temperature the highest frequency of sparkles occurred at a clock rate of 2MSPS (125kSPS) at 2.17 sparkle codes per billion conversions. The high and low temperature range are set by the specified minimum and maximum temperature range for the device. At high temperature (125°C) the maximum sparkle rate occurred at 0.329 ppb at VA set to 5 V. At low temperature (-55°C) the maximum sparkle rate occurred at 13.318 ppb at VA set to 5 V. These results are detailed later in this document, and are aligned with design expectations.

Different device configurations were also examined for sparkle code variations. One such configuration is continuous conversion mode, where the CS signal is held low throughout conversions instead of changing state. This configuration did not show any different frequency of sparkle occurrence than previously observed. The effect of multiplexing inputs was also considered, and also showed no variation from the expected. Through all data collection, the sparkle value has been fixed for the respective code transition being observed. No sparkle codes were observed to happen consecutively.

The ADC128S102-SP was released in 2008, with over a decade in the industry. It has a long flight history with numerous successful missions and is the most used ADC in the industry to date. The device has been used across multiple applications and a sparkle code has not been observed during normal operation; it has only been observed in rigorous test conditions. Even then, it is an extremely uncommon occurrence with a very precise controlled input condition, resulting in billions of conversions within that stringent set up for a sparkle code to occur. It is also important to note that since the device’s release, there have not been any changes to fabrication or test procedures of the device.

Although a sparkle code is a rare occurrence, it is a real possibility, and mitigation methods can be put in place to protect the system. Sparkle code occurrences resemble single event transient (SET) signatures, which will be safeguarded for applications with existing mitigation methods. This can result in minimum to no modifications in existing firmware. There are various simple mitigation methods that can be implemented depending on the application. Various methods to mitigate the error will be presented in this document. A *best two of three* approach is explained in this document and a pseudo code created by Texas Instruments is available in section Software Example.

## 3 Root Cause Analysis

A multi-step ADC architecture, such as a successive approximation register (SAR) ADC, converts a continuous analog input voltage to a set of digital bits through a binary search. The binary search utilizes a high-speed decision loop to determine each bit value. The binary search starts by setting the most significant bit and comparing this value with the input voltage to determine the correct value and then setting the next significant bit. It repeats this decision loop, bit-by-bit, to build the remaining bits of the converted result. This process can introduce a sparkle code in the ADC128S102QML-SP due to the logic in the ADC having insufficient settling time. The design of the ADC128S102QML-SP is susceptible to sparkle codes due to the meta-stability of the bit-decision flip-flops used to determine the output conversion result.

A simplified block diagram is shown in Figure 3-1. The time margin for settling the flip-flops on the ADC128S102QML-SP decreased from the commercial version of the device due to the added circuitry required for radiation hardening. Note that these design differences make the ADC128S102QML-SP susceptible to sparkle codes, whereas the catalog version of the device is not susceptible to sparkle codes. It is also possible that sparkle code occurrence may be exacerbated at higher power supplies and lower temperatures as in these fast corner conditions there is less margin. Due to the nature of a sparkle code lot to lot variation cannot be bounded, and should be expected to vary as any other parametric would.

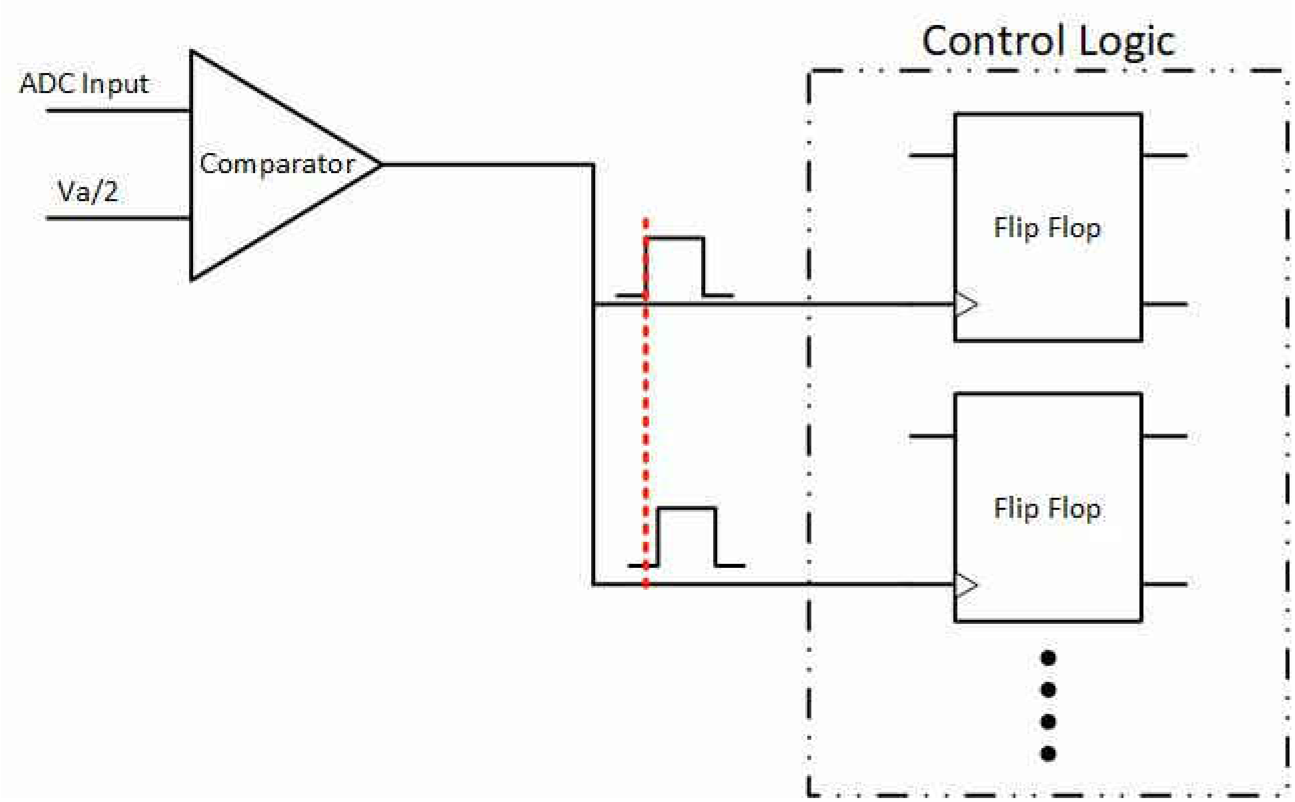

**Figure 3-1. Decision-Making Flip Flops Block Diagram**

The source of a sparkle code occurrence is inherent to the digital circuitry of the ADC, and depends on the output code transition the ADC is measuring. A sparkle code may occur if the input voltage results in any lower subset of the digital code to toggle near a boundary where the most significant bit (MSB) of that subset is *0* and least significant bits (LSB) are all *1*, or vice versa, where the most significant bit is *1* and the least significant bits are all *0*. The output states described can be clearly seen in Table 3-1, which lists the ADC output code transitions and the respective possible sparkle value at that code transition. To further explain, a sparkle code is possible when the input voltage is at a binary multiple of the analog power supply, which acts as the reference voltage for this device. As an example, when the full-scale range is 5 V, the voltage step size is 1.2207 mV, which is equivalent to a least significant bit (LSB) and a single code increment. Therefore, an input voltage between 2.5 V and 2.49938 V (2.5 – 1 LSB) is now between the code transition 0×800 and 0×7FF. This will cause the ADC to naturally toggle between *code* and *code–1* with equal probability based on quantization noise.

When the device is in this condition, a sparkle code is possible. Note that to achieve this condition, the input needs to have low noise and be equally within the code transition. If there is any noise, drift, or other sources of error, the output measurement could shift outside the code transition. If this occurs a sparkle code is no longer possible.

Each code transition will only sparkle to the respective listed sparkle value, no other value. The smallest sparkle delta as shown is 0×FF. The input voltage needs to be within half an LSB of this code transition for the device to output a sparkle code.

**Table 3-1. List of Input Ratio and Respective Sparkle Code Value**

| Vin/Vref Ratio | Code Transition | Sparkle Code |
|---|---|---|
| **1/16** | 0×0FF - 0×100 | 0×1FF |
| **2/16** | 0×1FF - 0×200 | 0×3FF |
| **3/16** | 0×2FF - 0×300 | 0×200 |
| **4/16** | 0×3FF - 0×400 | 0×7FF |
| **5/16** | 0×4FF - 0×500 | 0×5FF |
| **6/16** | 0×5FF - 0×600 | 0×400 |
| **7/16** | 0×6FF - 0×700 | 0×600 |
| **8/16** | 0×7FF - 0×800 | 0×FFF |
| **9/16** | 0×8FF - 0×900 | 0×9FF |
| **10/16** | 0×9FF - 0×A00 | 0×BFF |
| **11/16** | 0×AFF - 0×B00 | 0×A00 |
| **12/16** | 0×BFF - 0×C00 | 0×800 |
| **13/16** | 0×CFF - 0×D00 | 0×DFF |
| **14/16** | 0×DFF - 0×E00 | 0×C00 |
| **15/16** | 0×EFF - 0×F00 | 0×E00 |

# 4 Test Solution

To understand the sparkle behavior under multiple conditions a unique test solution had to be created to set the precise input conditions required to produce a sparkle code. The test solution has to be able to maintain the output code directly between the two codes listed under Code Transition in Table 3-1. During the test runs, data points before and after the detected sparkle code have to be collected. The test solution has to support collecting a significant number of conversions while monitoring the data for sparkle codes. The test solution also has to be robust, to repeat device behavior across different conditions.

The hardware created had to support the half LSB accuracy requirement. To accomplish this the Data Precision Multifunction Calibrator (DP8200) was used. The equipment has a 10 ppm reading + 1 ppm range accuracy, has 1 uV resolution, and can be remotely controlled was used as the input signal source. To improve the noise performance a low pass filter consisting of a 16Kohm resister and 1uF capacitor was added with a cut off frequency of 10Hz. The input drive circuit consists of an OPA320 and a charge bucket filter (R2, C2) to drive the sample and hold capacitor shown in Figure 4-1. The ADC128S102-SP uses the analog power supply (VA) as the reference voltage to measure the output. VA is driven by the TI REF60xx family of precision voltage references., which have a total noise performance of 5uV_RMS. The reference, amplifier and digital power are provided by an external power supply, Keysight E363A or HPE3611A. The hardware was designed to use a socket to be able to change devices quickly and isolate the device for temperature measurements without exposing the surrounding circuitry using a thermal stream.

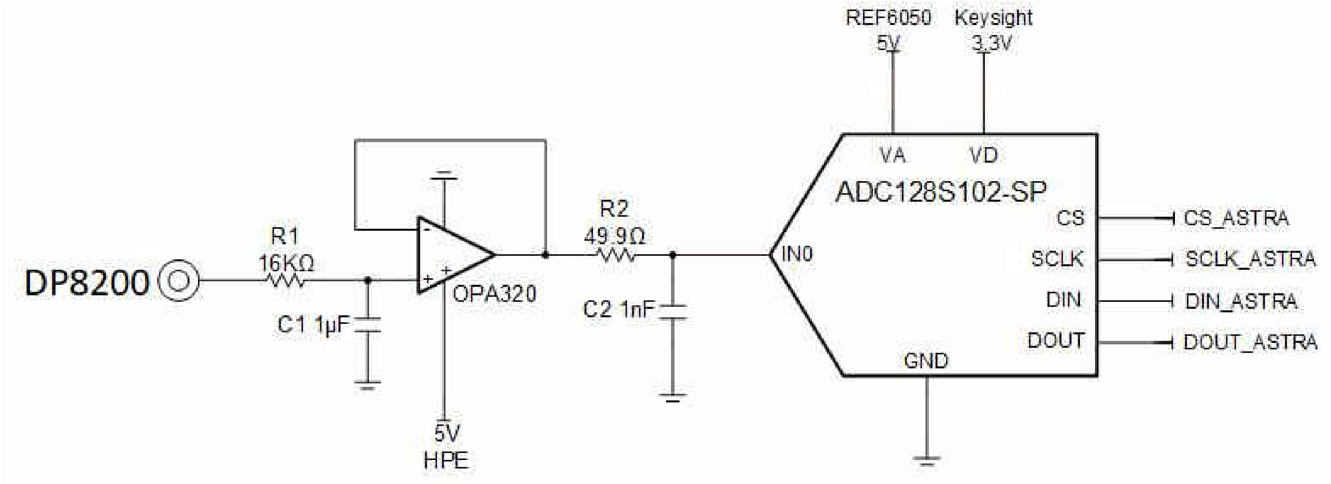

**Figure 4-1. Input Signal on Test Solution Schematic**

The ADC output is a digital representation of the analog input to ADC, thus to achieve a desired output code, the input voltage must be controlled to produce in the targeted output code. The algorithm used by the solution consist of a closed servo loop to control the input voltage based on the output code by controlling the DP8200 through GPIB cable, patent pending. The firmware can be thought of as two main functions, as shown in Figure 4-2, the Input Search Algorithm function and the Data Collection loop. The input search algorithm starts by bracketing the solution by a margin indicated by the user around the desired output code, and sets the input voltage to the midpoint of the bracket. The converter then completes 262,144 conversions and compares the average output code of this batch of conversions to the desired midpoint code transition. If the difference between the two values is not less than 0.1, then the input voltage is adjusted. To adjust the input, a half-interval search is used, if the error is greater than the desired value the upper bound is tightened by half and the same is done to the lower bound if the result is less than the desired value. The process continues as such until the average output code is within 0.1 codes of the midpoint code transition. This threshold is tighter than the targeted goal of half an LSB, which is the same as half a code. Referring to the previous example where the desired *code* and *code-1* are 0x800 and 0x7FF, Figure 4-2 demonstrates an expected histogram of the output codes once the input algorithm is complete.

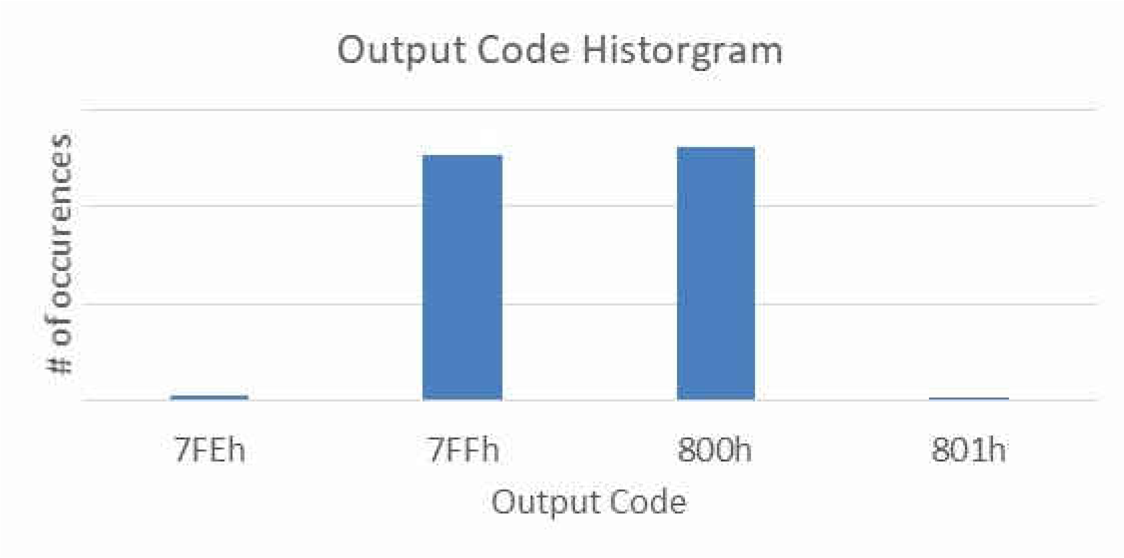

**Figure 4-2. Example Histogram of Output Code at Code Transition**

Once the input produces the required output, the firmware enters the data collection loop. The data collection loop continuously converts data, in batches containing 262,144 conversions, and compares each output code within the batch to the expected sparkle code value. If a sparkle value is not present, then the average output is then compared to an extended error threshold of 0.12 LSB. This is to avoid the firmware jumping in and out of the loop too frequently; when collecting data at temperature extrema, this error threshold was increased to 0.35 LSB. In both cases this threshold is still within the half LSB requirement. If a sparkle is present, the batch containing the conversion results with the sparkle code is saved, this provides the outputs code before and after a sparkle code occurrence. From this point, the firmware continues on the same loop explained until the specified number of loops desired.

The firmware provides an accurate input voltage signal through the DP8200 and can account for drift within the system. Over time and temperature, the voltage reference, amplifier, and precision input source can drift. The average output code is checked against the error threshold before collecting a new batch of conversion results. If the average output code becomes greater than the error threshold (0.12 room temperature and 0.35 over temperature), then the firmware will return to the input search algorithm to re-adjusted the input voltage to force the output code into the proper range for sparkle detection.

The firmware is run on an in-house platform, ASTRA, which uses C based coding language, and drives the SPI bus to and from the converter.

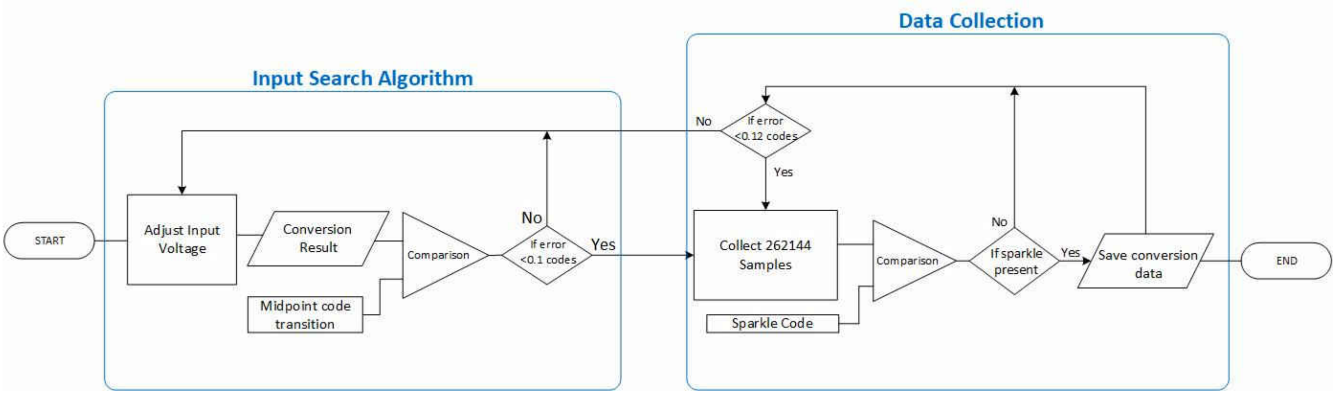

**Figure 4-3. Test Solution Firmware Flow Map**

## 5 Empirical Data and Results

The devices were tested at four different configurations at room temperature to observe for sparkle rate occurrence. The fist was a full-scale range sweep at the 15 input voltages listed in Table 5-1 with VA set to 5 V. The second observed response across clock rate. The third configured the devices in continuous conversion mode. The final configuration observed the effect of multiplexing between channels. All these tests were completed at room ambient temperature, with a VA at 5 V, and the digital supply was consistent at 3.3 V through all tests. Each configuration was observed for three test runs of equal size for each device, the results from all devices were combined to provide the following results.

The output code transitions were concluded to have an expected sparkle value by analysis of device design. The first test configuration was to verify the sparkle value across the full analog input range. The input voltage was set as a ratio of the full scale to result in the code transitions listed in Table 5-1. Three data collection runs consisting of 6.29 billion conversions per input ratio was completed, per device. The results are listed in Table 5-2, and range from no sparkle to 2 sparkles per billion conversions, at the expected sparkle value. From the table, it can be seen that all input ratios exhibit the expected sparkle, meaning all ratios are susceptible to a sparkle code. From this data, it can also be seen that certain input ratios experience higher sparkle rate than others. Two input ratios that the rest of this document will focus on, half scale (8/16) and near full scale (15/16) demonstrate this trend, as the half scale has a higher rate than the near full scale.

These first experiments were only conducted at clock rate set to 16MHz, and VA set to 5 V. The digital output code transition is what is being controlled, thus the true value of the voltage input is not considered here since the input signal will be adjusted to compensate for any changes in VA. Also, the corner condition at higher VA levels is expected to demonstrate worse case.

**Table 5-1. Configuration One Results: Sparkle Rate Across Full Input Range**

| # conversions per input<br>6,291,456,000 | Sparkle Code Occurrence<br>(per billion conversions) | | |
|---|---|---|---|
| **Input Ratio** | **Run 1** | **Run 2** | **Run 3** |
| **1/16** | 1.113 | 1.51 | 1.03 |
| **2/16** | 1.113 | 0.556 | 0.794 |
| **3/16** | 0.318 | 0.238 | 0.238 |
| **4/16** | 1.748 | 1.192 | 1.98 |
| **5/16** | 0.795 | 0.874 | 0.556 |
| **6/16** | 0.238 | 0.079 | 0.079 |
| **7/16** | 0.318 | 0.238 | 0.317 |
| **8/16** | 0.477 | 1.19 | 1.11 |
| **9/16** | 0 | 0.238 | 0.079 |
| **10/16** | 0.159 | 0.158 | 0.079 |
| **11/16** | 0.159 | 0.238 | 0.158 |
| **12/16** | 0 | 0 | 0.158 |
| **13/16** | 0.159 | 0.397 | 0.397 |
| **14/16** | 0 | 0.079 | 0 |
| **15/16** | 0.238 | 0.317 | 0.158 |

The second test configuration focused on clock rate; the clock rate directly controls the sample rate of the device. The sampling rate is found by diving the clock rate by 16. The device analog power supply is set to 5V, and only the midscale input ratio (8/16) was observed for all tests. The clock rates observed are 2MHz, 8MHz, and 16Mhz, correspond to 125Ksps, 500Ksps, and 1Msps sampling rate respectively. Each data collection run consisted of over 15.2 billion conversions per clock rate, per device. All the results were combined and displayed on Table 5-2. The maximum sparkle rate occurred at 2.17 per billion conversion at 2MHz clock rate, and the least sparkle occurrence at 0.953 conversions per billion at 16 MHz clock rate. It can be inferred from the table that lower clock rates have a slightly higher rate of sparkle occurrence, but not significant enough to deter a sampling rate over another.

**Table 5-2. Configuration Two Results: Sparkle Rate Across Clock Rate**

| # conversions per clock rate<br>15,204,352,000 | Sparkle Code Occurrence<br>(per billion conversions) | | |
|---|---|---|---|
| **Clock Rate** | **Run 1** | **Run 2** | **Run 3** |
| **2 MHz** | 1.94 | 2.17 | 2.006 |
| **8 MHz** | 1.44 | 1.15 | 1.57 |
| **16 MHz** | 0.953 | 1.24 | 1.47 |

The third test configuration focused on operating the device in continuous conversion mode. To operate the device in continuous conversion mode the CS signal needs to be held low across consecutive 16-clock-pulse long frames. The timing diagram seen in Figure 5-1 can also be found the ADC128S102QML-SP data sheet, and shows how to operate the device in continuous conversion mode. For this test, we only measured at a single channel consecutively.

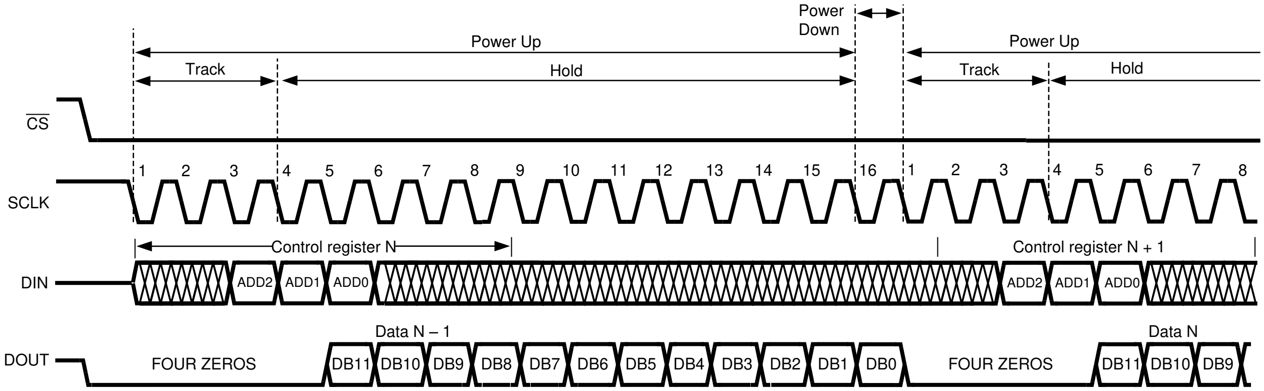

**Figure 5-1. Timing Diagram for Continuous Conversion Mode**

The device analog power supply is set to 5V, and the input ratio used are midscale input ratio (8/16) and near full scale ratio (15/16). The clock rate was set to 16Mhz for all data collection runs. Three data collection runs consisting of 6.29 billion conversions per input ratio was completed, per device. The combined results are available in Table 5-3. From the results, it can be seen that a low sparkle occurrence of 0.158 sparkle per billion conversion was seen. It is important to note that the sample size for this configuration is less than half of the previous, second configuration. Continuous mode did not show a substantial difference in results than expected, although it seemed to show a slight lower sparkle occurrence but not significant enough to deter a sampling mode over the other.

**Table 5-3. Configuration Three Results: Sparkle Rate with Continuous Conversion Mode**

| # conversions per input<br>6,291,456,000 | Sparkle Code Occurrence<br>(per billion conversions) | | |
|---|---|---|---|
| **Input Ratio** | **Run 1** | **Run 2** | **Run 3** |
| **8/16** | 0.158 | 0.158 | 0.158 |
| **15/16** | 0.158 | 0 | 0 |

The final device configuration tested was to determine if multiplexing between channels would have any effects on sparkle occurrence. The device was set to analog power supply of 5 V, and used a slower clock rate of 2MHz to eliminate any potential settling errors. Channel 0 was connected to the DP8200 to be driven by the firmware solution and channel 1 was at a DC non-sparkle code transition value, this input was not driven by the test solution. The device would then mux between the two channels monitoring for a sparkle code; the only recorded sparkle value occurred in channel 0 which was driven by the DP8200. Each data collection run consisted of 6.29 billion conversions, per device. The results are listed in Table 5-4, and are in line with expected results, falling within 1 to 2 sparkle codes per billion conversions. It is important to note that at this clock rate, the results coincide with previous results at this same clock rate.

Table 5-4. Configuration Four Results: Sparkle Rate with Mux-ed Inputs Channels

| # conversions per input<br>6,291,456,000 | Sparkle Code Occurrence<br>(per billion conversions) | | |
|---|---|---|---|
| Input Ratio | Run 1 | Run 2 | Run 3 |
| 8/16 | 1.58 | 1.74 | 1.82 |

Once an understanding of sparkle rate was achieved through the previous tests at room ambient temperature, the devices were tested at each temperature extrema. The devices were tested at -55°C and at 125°C, the limits of the device operating temperature. The analog power supply was also observed in this test configuration at 5 V and at 3.3 V. The sample size was decreased from previous test due to temperature exposure.

The first test configuration set the device analog power supply to 5 V, driven by the REF6050; the input ratio used are midscale input ratio (8/16) and near full scale ratio (15/16). The clock rate was set to 16Mhz for all data collection runs. The results were then combined for all devices tested to solve for the relation to per billion conversions and are listed in Table 5-5 and Table 5-6 for temperatures -55°C and at 125°C respectively.

From the results, it is evident the devices experience a higher rate of sparkle code per billion conversion at -55°C, while at 125°C the results show similar or lower sparkle code occurrence than previous results. This is aligned with corner condition performance expectations. It is also evident that the half scale input has a higher sparkle rate than the near full scale as expected.

Table 5-5. Sparkle Rate at -55°C with VA set to 5 V

| | Sparkle Code Occurrence<br>(per billion conversions) | | |
|---|---|---|---|
| Input Ratio | Run 1 | Run 2 | Run 3 |
| **8/16** | 8.55 | 12.989 | 13.318 |
| **15/16** | 0.53 | 3.179 | 2.119 |

Table 5-6. Sparkle Rate at 125°C with VA set to 5 V

| | Sparkle Code Occurrence<br>(per billion conversions) | | |
|---|---|---|---|
| Input Ratio | Run 1 | Run 2 | Run 3 |
| **8/16** | 0.329 | 0.329 | 0.164 |
| **15/16** | 0 | 0 | 0 |

The next test configuration solely changed the analog power supply to 3.3 V driven by the REF6033; recall that the analog power supply is also the reference in this device. All other parameters remained the same as the previous configuration. Results for temperatures -55°C and 125°C are listed in Table 5-7 and Table 5-8 respectively.

At lower analog power supply, the sparkle code rate greatly decreased, as seen in the following tables, and is aligned with corner condition performance expectations. At -55°C a max sparkle rate seen is 0.53 per billion conversion, at half scale, which is aligned with known trends. At -125°C a sparkle code occurrence was nor observed across any of the devices.

Table 5-7. Sparkle Rate at -55°C with VA set to 3.3 V

| | Sparkle Code Occurrence<br>(per billion conversions) | | |
|---|---|---|---|
| Input Ratio | Run 1 | Run 2 | Run 3 |
| **8/16** | 0 | 0.53 | 0 |
| **15/16** | 0 | 0 | 0 |

**Table 5-8. Sparkle Rate at 125°C with VA set to 3.3 V**

| | Sparkle Code Occurrence (per billion conversions) | | |
|---|---|---|---|
| **Input Ratio** | **Run 1** | **Run 2** | **Run 3** |
| **8/16** | 0 | 0 | 0 |
| **15/16** | 0 | 0 | 0 |

## 6 Mitigation Methods

The intermittent occurrence of the sparkle code allows for simple mitigation methods to safeguard the system from unnecessarily responding to a sparkle occurrence. Due to the nature of the sparkle code, any existing hardware does not need to be modified; sparkle code occurrence is inherently a digital response, and thus firmware focused method can address a sparkle code occurrence in existing applications. A sparkle code occurrence can resemble a single event transient (SET) considering how the output can suddenly be 0x0FF different from the expected measurement. If systems already have existing methods to responds to SET signatures, it is possible that no further mitigation methods are necessary. The best mitigation method will depend on the end application and device use case, thus there are many methods to mitigate the sparkle code.

The method highlighted in this section is a short firmware algorithm that eliminates an outlier output code, similar to a simple average. This simple firmware algorithm addresses any sparkle occurrence with a *best out of 3* approach. This technique requires three consecutive measurements, which are then averaged together. The three ADC measurements are then compared to the averaged result. The output code with the greatest deviation from the average is then eliminated. The final two measurements are then averaged together to provide a more accurate result. The system can move forward using this value to make any further decisions. The end user can also decide to forgo the second averaging and continue with the two remaining output measurements, instead of following the presented technique. The initial technique does limit the ability to measure single code outputs, but provides a safe approach with only limiting response to three measurements. This approach provides a quicker response time opposed to longer complex alternatives. In section Software Example, a pseudo code is available to implement this technique.

To protect the system from responding to a sparkle code, several options exists to post process the data. If the system allows for a longer response time, taking the average of a large number of samples decreases the sparkle code weight in the final result, averaging out the sparkle code. This approach can be extended with a digital filter. A digital filter can shape the frequency response and lower noise, at the possible expense of a longer response time. The digital filter would essentially screen all output measurements and eliminate any sparkle codes present.

Even though this can be addressed through firmware, there are hardware mitigation methods possible for new projects as well. If there is available printed circuit board space, redundancy in the hardware can be implemented. A second ADC can be introduced and sample the same input, the system can then compare output measurements and respond when needed only if both ADC outputs match.

## Software Example

```
Mitigation Pseudo Code, best of three samples //
****************************************************************
//Script Name : Best of 3 conversion samples
//Author : Art Kay
//Description : compare three samples and discard the furthers outlier of the three
// Copyright (c) 2021 Texas Instrument Incorporated //
****************************************************************
// Collect 3 samples
double Meas1 = 1000; //sample 1
double Meas2 = 1001; //sample 2
double Meas3 = 1;    //sample 3
//Define formula variables
double nAVG = 0.0;   //define average variable
double Result = 0.0; //define result variable
double Test1 = 0.0;  //define variable
double Test2 = 0.0;  //define variable
double Test3 = 0.0;  //define variable
//Calculate average sample
nAVG = (Meas1 + Meas2 + Meas3)/3; //initial average for comparison
//Determine which sample has the largest deviation from the average value
Test1 = Abs(Meas1 - nAVG);
Test2 = Abs(Meas2 - nAVG);
Test3 = Abs(Meas3 - nAVG);
//Re-calculate the average value, ignoring the sample with the largest deviation
if(Test1 >= Test2 && Test1 >= Test3)
{
 Result = (Meas2 + Meas3)/2;
}
elseif(Test2 >= Test1 && Test2 >= Test3)
{
 Result = (Meas1 + Meas3)/2;
} elseif(Test3 >= Test1 && Test3 >= Test2)
{
 Result = (Meas1 + Meas2)/2;
}
end;
```

# 7 Conclusion

The ADC128S102QML-SP can result in specific sparkle code values depending on the output code transition, but this result is intermittent and is in the order of parts per billion. Across multiple configurations and the device's temperature range the sparkle code maintained the infrequent rate of occurrence. To address a possible sparkle code, the system's requirements and limitations must be considered to select the best fitting method. There are multiple mitigation methods that can be implemented in firmware as well as hardware, including a simple *best of three* approach at the ADC output. The ADC128S102QML-SP has over a decade of flight history and great success the aviation and space industry; it is the most widely used ADC in the industry.

# 8 Revision History

| Changes from Revision * (May 2021) to Revision A (June 2022) | Page |
|---|---|
| • Updated the *List of Input Ratio and Respective Sparkle Code Value* table | 4 |

# IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATA SHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES "AS IS" AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, regulatory or other requirements.

These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI's products are provided subject to TI's Terms of Sale or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI's provision of these resources does not expand or otherwise alter TI's applicable warranties or warranty disclaimers for TI products.

TI objects to and rejects any additional or different terms you may have proposed.

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