User's Guide

Q&A Watchdog Overview and Configuration

Eric Schott *Transceiver Interface*

ABSTRACT

This user's guide presents the function and use of the Q&A watchdog feature, using TCAN1146-Q1 as an example device. The fundamentals of a watchdog are explained, as well as three common implementations of the feature. The watchdog function on the TCAN1146-Q1 is used to present example configurations to further explain how it is used.

Table of Contents

List of Figures

List of Tables

Trademarks

All trademarks are the property of their respective owners.

1 General Description of Watchdog

Watchdog features are common functions in the electronics industry used to increase system reliability. A watchdog is a timer meant to verify that the controller of a device is working correctly. This is done by requiring the controller of a device to periodically update or reset a timer in the controlled device based upon a specific timing sequence. If this action is not executed correctly, an interrupt or timeout signal is generated and depending on the system design, some kind of corrective action takes place. In systems where human intervention is not easy, possible, or cannot take place quickly enough, the system autonomously fixes any issues with its main controller by forcing it to reset. This article further explains what a Q&A watchdog is, how it works, and what considerations need to be made when this feature is being configured. This document uses the TCAN1146-Q1 from Texas Instruments as an example and examines the features, sequences, and registers.

2 Timeout Watchdog

The timeout watchdog is the simplest implementation of the watchdog function. The general concept goes as follows: a timer is started as soon as the first watchdog trigger is sent, and that same trigger must be sent within the configured time limit, otherwise an error is asserted. In most cases, an error counter is incremented when the timer runs out before a trigger occurs. When this counter reaches a certain value, a fault (interrupt) is sent to the controller to initiate the diagnostic function or reset mechanisms.

3 Window Watchdog

The window watchdog is similar to the timeout watchdog, except the timer is split into an opened and closed window. As the name implies, the open window is when the watchdog trigger can be sent and accepted, and the closed window is the time range when the watchdog trigger cannot be sent. If the trigger is sent during the closed window, or if the open window times out without the trigger being sent, the error counter is incremented. The idea behind this version of a watchdog timer is that the microcontroller has to be a bit more precise, and avoids a loop-lock situation, where the microcontroller can still service a timeout watchdog while being stuck in a never-ending loop.

4 Q&A Watchdog

The question and answer (Q&A) watchdog takes aspects from both timeout and window watchdog types with the added complexity of posing a query to the controller. In this method, the controller is expected to periodically service the watchdog by presenting the answers to specific questions. These questions can either be requested from the watchdog device, be based on a previous answer, or be a part of a predetermined sequence. The answers to each question must be provided in the correct order and within a certain time window. The questions are simple math functions or bit-shifting operations that require the controller to be active and responsive to dynamic conditions rather than simply meeting a timing requirement. If the controller fails to respond to a question within its window, responds to any message out of order, or responds with an incorrect answer, an error is asserted.

5 Example Q&A Watchdog With TCAN1146

In order to look at how a Q&A watchdog operates and to understand the functions of the Q&A watchdog timer, we will consider the feature's implementation in TCAN1146 . This device is a CAN transceiver with a serial peripheral interface (SPI) and a watchdog feature. When the watchdog timer in this device is enabled, it can be configured in one of three modes: timeout, window, or Q&A. This document focuses on a Q&A implementation.

Note

Because Q&A watchdogs require elements from timeout and window watchdog types, it is common for devices that feature Q&A to allow these simpler watchdog timer configurations as well.

6 Watchdog Configuration

When initializing the TCAN1146, various configurable characteristics of the watchdog must be selected and enabled. This includes the watchdog type (Q&A in this example), window timing (which defines how long each response window is and ultimately the length of the watchdog cycle), error counter threshold (this defines how many errors need to accumulate before a watchdog event triggers), a trigger action (the action that occurs once the error count exceeds the defined threshold), and question generation type, polynomial, and seed (these characterize how answers are to be calculated based on the question values). The table below lists the registers in TCAN1146 where these characteristics may be configured. The device datasheet contains a description of each register and what bits map to different functions.

Table 6-1. List of TCAN1146 Watchdog-Related Registers

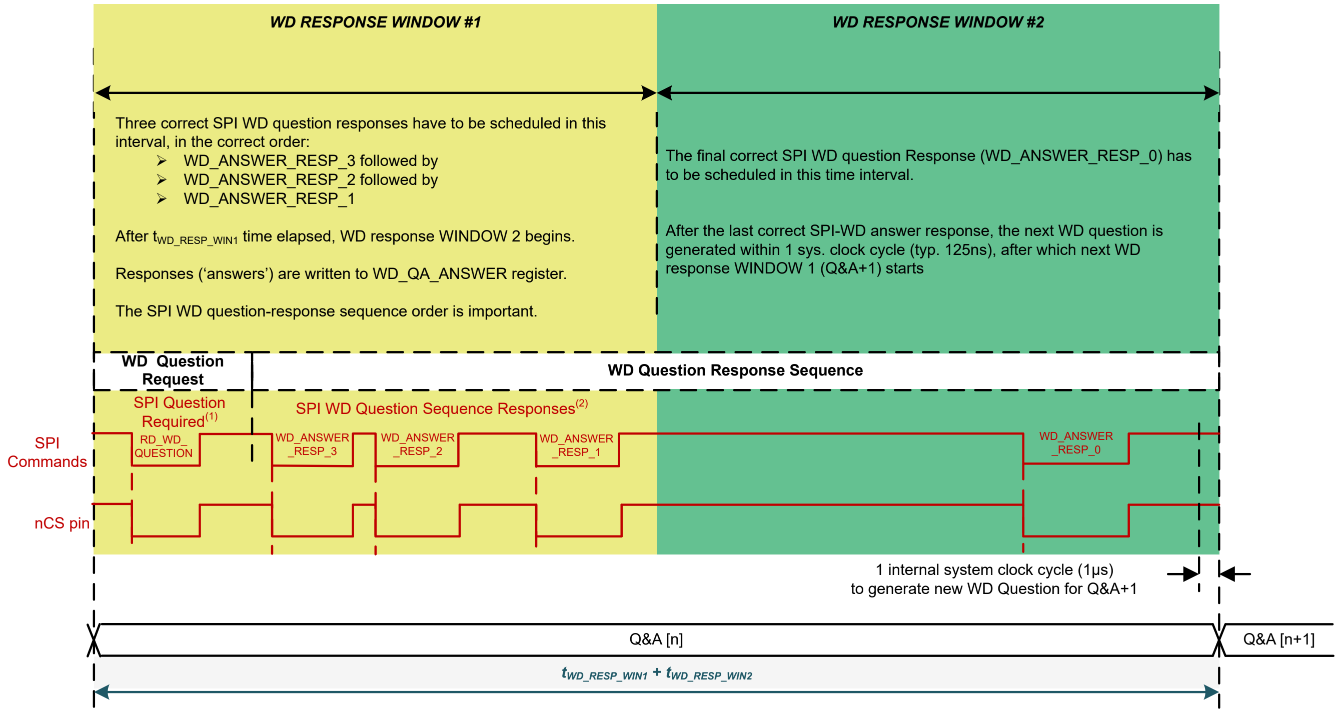
Register Address	Register Name	Description
0x13	WD_CONFIG_1	Watchdog configuration and action in event of a failure
0x14	WD_CONFIG_2	Sets the time of the window, and shows current error counter value
0x15	WD_INPUT_TRIG	Register to reset or start the watchdog
0x16	WD_RST_PULSE	Reset pulse width in event of watchdog failure
0x2D	WD_QA_CONFIG	Configuration related to the QA configuration
0x2E	WD_QA_ANSWER	Register for writing the calculated answers
0x2F	WD_AQ_QUESTION	Reading the current QA question

7 Watchdog Services

The Q&A watchdog must be serviced once every watchdog cycle. This time is defined by the window time configured in the WD_TIMER and WD_PRE register bits. The watchdog cycle is split into two response windows that are each 50% of the watchdog cycle time: WD Response Window #1 and WD Response Window #2. During the first window, the question is read by the controller[1] and the first three answers are sent back. The controller then waits before sending the fourth and final answer in the second window. At the end of the second window, a new watchdog cycle begins and the process repeats.

Note

When using the Q&A watchdog, it is recommended to use window times greater than 64 ms due to the need for several bytes of SPI to be used for each watchdog Q&A event.



Three correct SPI WD question responses have to be scheduled in this interval, in the correct order:

- WD_ANSWER_RESP_3 followed by
- WD_ANSWER_RESP_2 followed by
- WD_ANSWER_RESP_1

After $t_{WD\_RESP\_WIN1}$ time elapsed, WD response WINDOW 2 begins.

Responses ('answers') are written to WD_QA_ANSWER register.

The SPI WD question-response sequence order is important.

The final correct SPI WD question Response (WD_ANSWER_RESP_0) has to be scheduled in this time interval.

After the last correct SPI-WD answer response, the next WD question is generated within 1 sys. clock cycle (typ. 125ns), after which next WD response WINDOW 1 (Q&A+1) starts

Figure 7-1. Timing and Sequence for Watchdog Q&A Multi-Answer Mode

[1] Reading the question from the watchdog is not required if the answers are already known. The first transaction during this window can be the response sequence.

7.1 Watchdog Good Event

Each time a successful watchdog cycle is completed, the error counter decrements by one. This means that a few failed cycles (depending on the configured error counter threshold) can be tolerated before a watchdog trigger occurs. If some failures occur before the controller recovers, but not enough to surpass the threshold, the subsequent successful cycles can return the counter to zero without the need for a trigger or reset.

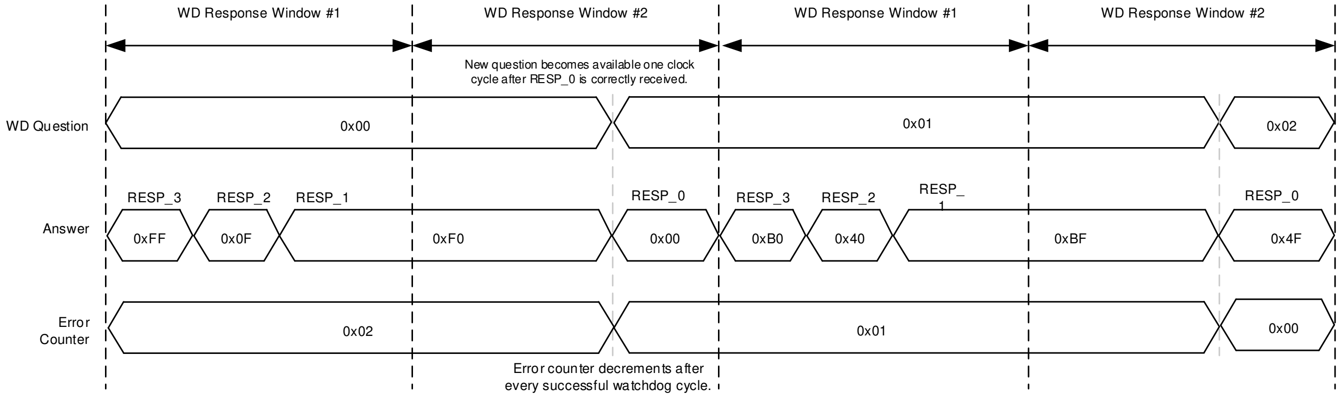


Figure 7-2. Successful Watchdog Servicing Sequence Example

7.2 Watchdog Incorrect Answer

A failed cycle occurs if the controller does any of the following; fails to request a question before a timeout, responds to any message out of order or outside of the specified window, or responds with an incorrect answer. When a failed cycle occurs, the error counter increments. Once the error counter surpasses the configured threshold, the device triggers a watchdog event. The TCAN1046 can be configured to pulse the power enable line (INH) to reset the local node and controller and set a watchdog interrupt, or to only set a watchdog interrupt, allowing another method to recover the system. An incorrect answer in any response from the controller causes the cycle to fail and the error counter to increment. Such an event may indicate that the controller is not properly responding to outside signals and must be reset.

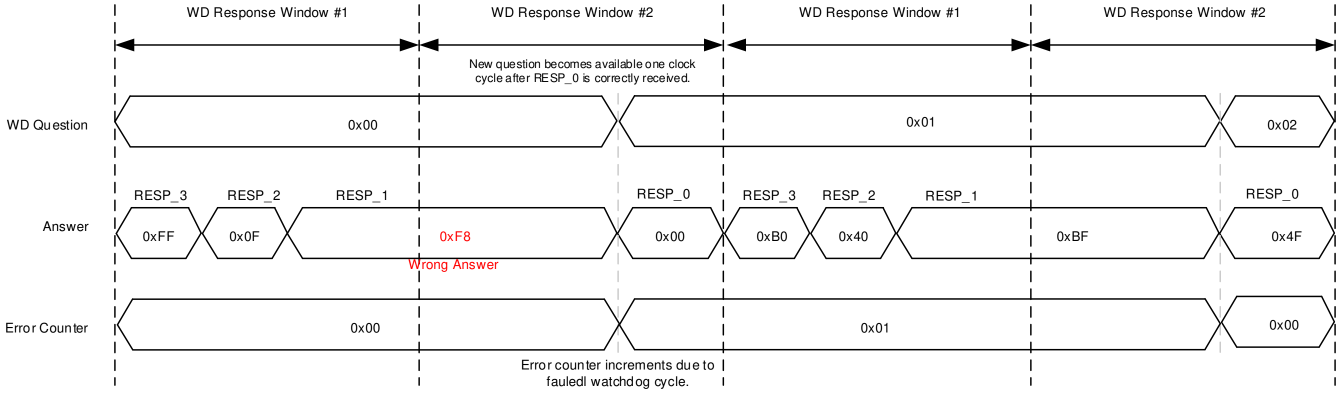


Figure 7-3. Example of a Failed Cycle Due to an Incorrect Response

7.3 Watchdog Out of Timing

When the controller writes an answer during the wrong response window, this causes the cycle to fail and the watchdog increments the error counter. This is the same check that a window watchdog conducts and aims to prevent timing errors from compounding in a system, allowing a reset if the error becomes large and frequent enough.

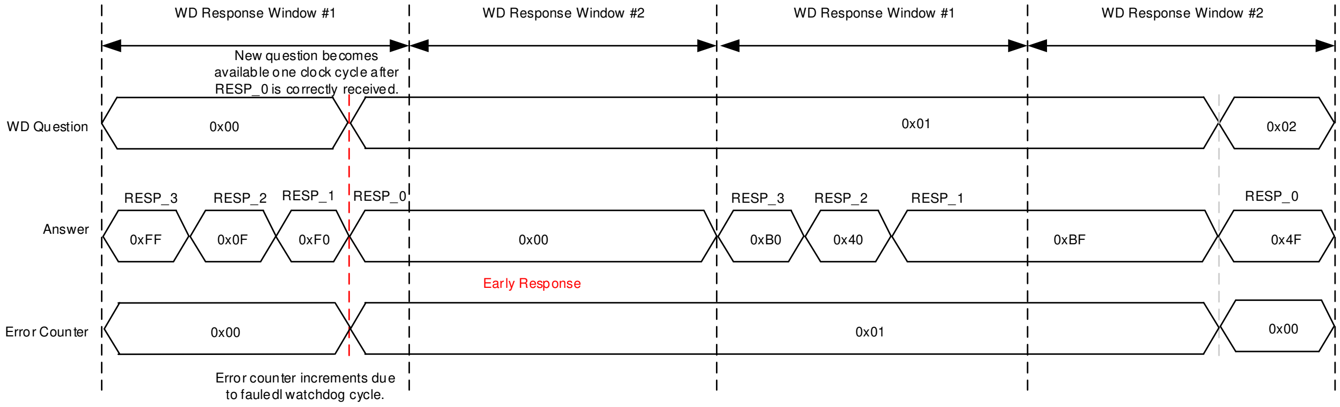


Figure 7-4. Example of a Failed Cycle Due to a Missed Response Window

7.4 Watchdog No Response

If the controller does not respond to the watchdog with the correct number of answers before the end of a timing window, the cycle fails and the error counter is incremented. This echoes the functionality of the timeout watchdog and may indicate that the controller is hung up on an internal loop.

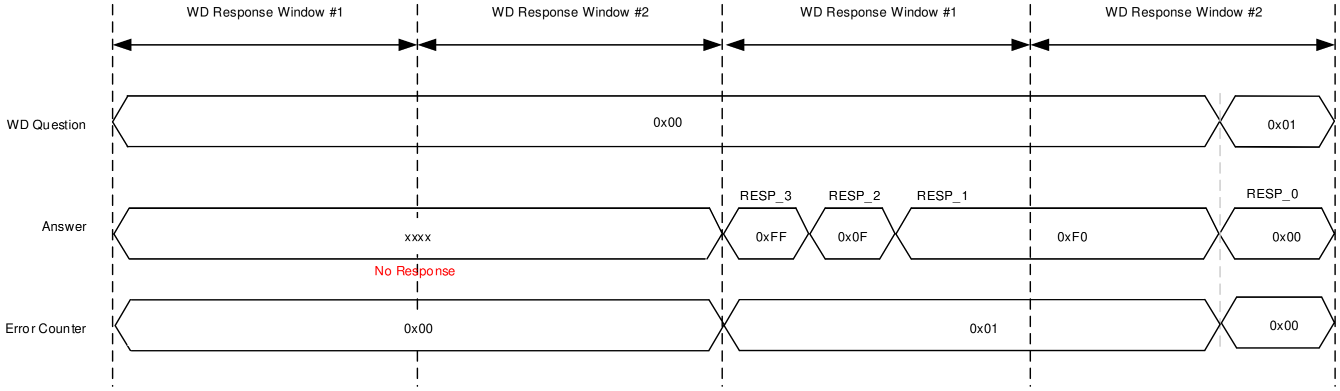


Figure 7-5. Example of a Failed Cycle Due to the Controller Failing to Respond Before the End of the Timing Window

8 Summary

Q&A watchdogs combine the features used in timeout and window watchdogs with the added requirement of state-based responses. Systems implementing this fail-safe design benefit from increased reliability and assure that the controller is continuously capable of moderately complex external interfacing and internal operational consistency.

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATASHEETS), 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, 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 (https:www.ti.com/legal/termsofsale.html) 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.

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