APPLICATION NOTE 7341

DISABLE THE WATCHDOG TIMER DURING SYSTEM REBOOT

By: Suryash Rai

Abstract: In most of the microprocessor applications where a watchdog supervisor, such as the MAX6369 is used to monitor system activity, it may be necessary to disable the watchdog during system reboot. This is particularly true when software boot times exceed the watchdog time-out period. This application note describes a circuit that can be used to selectively disable the watchdog timer

Introduction

Watchdog timers are the fundamental parts of modern embedded systems. A watchdog timer continuously watches the execution of code and resets the system if the software hangs or no longer executes the correct sequence of code.

The Microcontroller Unit (MCU) starts the bootup process when it is powered up, which can take several milliseconds to a few seconds. The microcontroller performs system initialization and other housekeeping activities during the bootup process. The MCU, while running the bootup sequence, does not expect any interruptions or RESET signal from the external watchdog timer monitoring the MCU. This is the main reason watchdog timers have long power-on or startup delays.

The watchdog timer keeps asserting the watchdog output (WDO) before the bootup finishes in applications where the bootup process is longer than the watchdog timeout period. This creates a deadlock and the MCU hangs forever.

This application note explains how the MAX6369–MAX6374 family can eliminate this deadlock.

Application Circuit and Description

The MAX6369–MAX6374 are a family of pin-selectable watchdog timers that supervise the MCU activity, and signal when the MCU is stuck in a loop or does not execute code. The microprocessor must repeatedly toggle the watchdog input (WDI) during normal operation before the selected watchdog timeout period elapses to demonstrate the system is processing code properly. The MAX6369–MAX6374 assert a watchdog output (WDO) if the MCU does not provide a valid watchdog input transition before the timeout period expires. The watchdog output pulse can be used to reset the MCU or interrupt the system to warn of any processing errors. **Table 1** shows the different timing options provided by the SET pin configuration of the MAX6369–MAX6374.

Table 1. Minimum Timeout Setting

LOGIC INPUT			MAX6369/MAX6370	MAX6371/MAX6372	MAX6373/MAX6374	
SET2	**SET1**	**SET0**	**t_{DELAY}, t_{WD}**	**t_{DELAY} = 60s, t_{WD}**	**t_{DELAY}**	**t_{WD}**
0	0	0	1ms	1ms	3ms	3ms
0	0	1	10ms	3ms	3s	3s
0	1	0	30ms	10ms	60s	1s
0	1	1	Disable	Disable	Disable	Disable
1	0	0	100ms	100ms	200µs	30µs
1	0	1	1s	300ms	First Edge	1s
1	1	0	10s	3s	First Edge	10s
1	1	1	60s	60s	60s	10s

Figure 1 is the application diagram of the MAX6369 interface with the MCU.

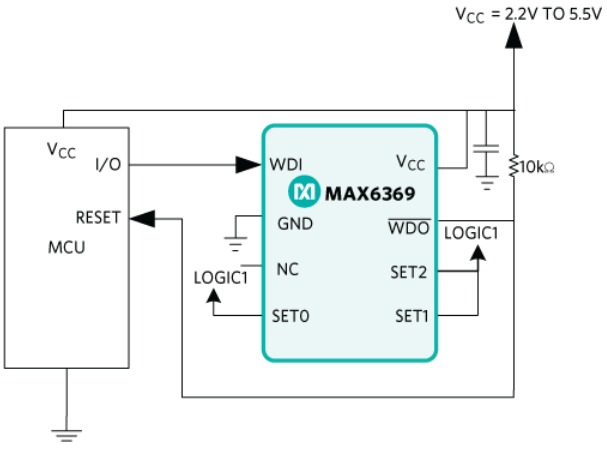

Figure 1. Application diagram of MAX6369.

Benefits and Features

- Precision watchdog timer for critical µP applications

- Pin-selectable watchdog timeout periods
- Pin-selectable watchdog startup delay periods
- Ability to change watchdog timing characteristics without power cycling
- Open-drain or push-pull pulsed active-low watchdog out
- Watchdog timer disable feature
- Operating voltage of +2.5V to +5.5V
- Low-supply current of 8µA
- No external components required
- Miniature 8-pin SOT23 package

Figure 2 is the timing diagram of the MAX6369–MAX6374. The MCU starts its bootup after the device power-up. The MAX6369 ignores any WDI violation for t_{SETUP} + t_{DELAY}, which is longer than the bootup process. The MCU, after the bootup process, starts toggling the WDI pulse before the t_{WD} time expires.

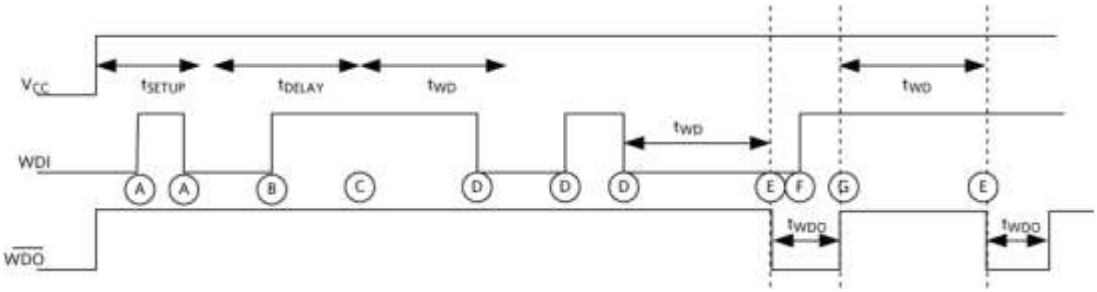

Figure 2. Timing diagram of MAX6369.

The following are the different key timing instances during the operation:

a. Transitions on WDI ignored during setup delay.
b. Transition(s) on WDI ignored during startup delay.
c. Watchdog timer starts after startup delay and WDO deasserts.
d. Transition occurs before watchdog timeout period (t_{WD}).
e. Watchdog timer clears and starts timer again. Watchdog times out (> t_{WD}) and WDO asserts.
f. Transitions on WDI ignored when WDO asserts.
g. Watchdog timer starts after WDO deasserts.

Deadlock Condition During System Reboot

The MAX6369-MAX6374 starts monitoring the MCU activity after the bootup process is finished.The watchdog timer detects the MCU watchdog fault if the MCU is unable to toggle the WDI pulse before the t_{WD} time expires (**Figure 2**). The watchdog fault reboots the MCU. If the system reboot time is more than tWD, the MAX6369 toggles the WDO repeatedly before the reboot process finishes. This hangs the MCU forever. **Figure 3** shows a similar deadlock condition.

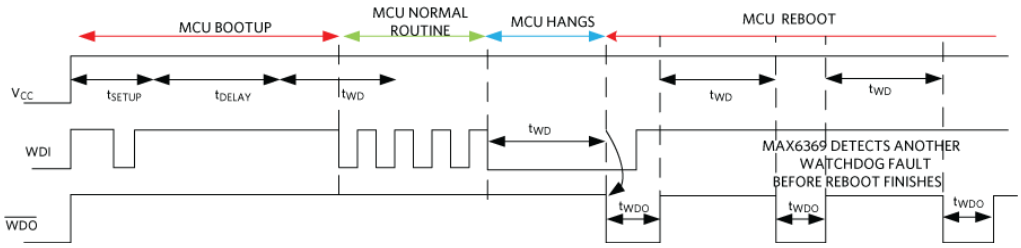

Figure 3. Deadlock condition in the MCU.

The deadlock problem in Figure 3 is solved by providing extra setup delay (t_{DELAY}) to the MAX6369–MAX6374 every time it detects a watchdog fault.

The MAX6369–MAX6374 provides the 3-set pin to achieve the desired performance. The MAX6369–MAX6374 allows the user to change the timing setting dynamically. If the set pin configuration is changed after the WDO is asserted, the previous setting is allowed to finish. The characteristics of the new setting are assumed after the WDO is deasserted, and the MAX6369–MAX6374 enters into a new startup phase. **Figure 4** is the timing diagram of MAX6369–MAX6374, where the set pin configuration is changed after the WDO is asserted.

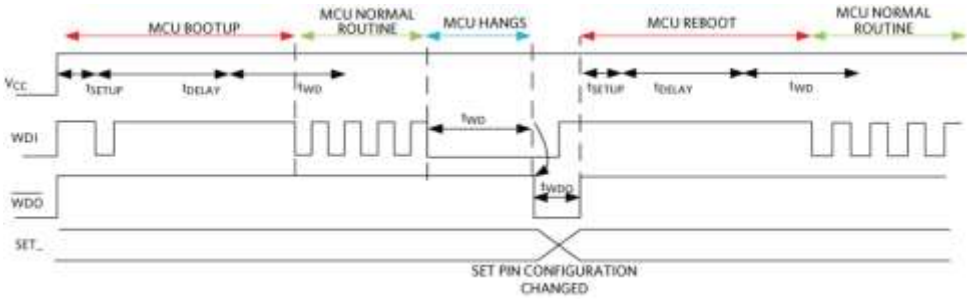

Figure 4. Changing the SET pin configuration when WDO is asserted.

Figure 5 shows the application circuit diagram used to get the timing diagram in **Figure 4**. The SET1 logic pin is connected to the WDO pin in the following schematic.

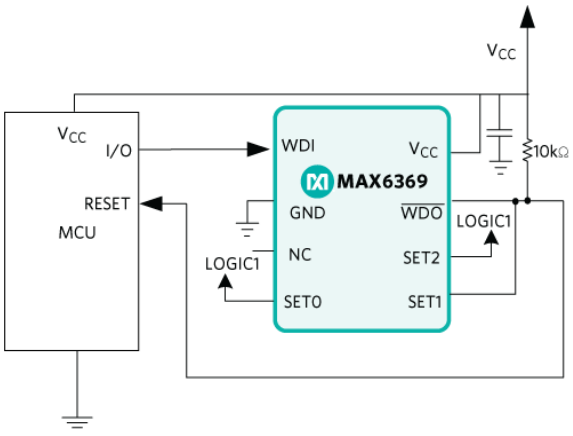

Figure 5. Application circuit to disable the watchdog timer during the reboot.

The initial setting of the MAX6396 is SET0 = 1, SET1 = 1, and SET2 = 1 after the device power-up. This sets the t_{DELAY} and t_{WD} to 60s. If the MAX6369 detects the watchdog fault. The WDO asserts and also changes the SET pin setting to SET0 = 1, SET1 = 0, and SET2 = 1. This setting lasts only for t_{WDO} (100ms). The WDO deasserts and the SET pin configuration changes to its initial setting of SET0 = 1, SET1 = 1, and SET2 = 1. The transition in the SET pin initiates the new setup phase, which includes the t_{SETUP}, t_{DELAY}, and t_{WD}. The MAX6369 does not assert any watchdog fault during the MCU reboot in the new circuit arrangement. The system restarts normally after the watchdog fault. This solution is only valid if at least one of the SET pins is chosen as logic 1. The SET pin cannot be connected to the WDO output if all the SET pins are logic 0.

Figure 6 shows the MAX6369 with the circuit connection in Figure 1, where SET0 = V_{CC} (logic 1), SET1 = V_{CC} (logic 1), and SET2 = V_{CC} (logic 1). The WDI is connected to the V_{CC} to observe the MAX6369 watchdog faults. The device waits for the t_{SETUP}, t_{DELAY}, and t_{WD} to assert the WDO pulse after power-up. It keeps toggling the WDO pulse after t_{WD}.

Figure 6. MAX6369 with SET0 = V_{CC}, SET1 = V_{CC}, and SET2 = V_{CC}.

Figure 7 shows the MAX6369 with the circuit connection in Figure 5, where SET0 = VCC (logic 1), SET1 = WDO, and SET2 = VCC (logic 1). The device starts the new startup phase every time the MAX6369 detects a watchdog fault.

Figure 7. MAX6969 with SET0 = V_{CC}, SET1 = WDO, and SET2 = V_{CC}

Summary

The MAX6369–MAX6374 family of watchdog timer ICs can monitor the MCU timing misbehavior and solve the deadlock condition typically seen in embedded systems without additional discrete components.

Related Parts	
MAX6369	Pin-Selectable Watchdog Timers
MAX6370	Pin-Selectable Watchdog Timers
MAX6371	Pin-Selectable Watchdog Timers
MAX6372	Pin-Selectable Watchdog Timers
MAX6373	Pin-Selectable Watchdog Timers
MAX6374	Pin-Selectable Watchdog Timers