Keywords: Watchdog Timer, Reset, Timeout, Code, Toggle, Software error, Reliability

APPLICATION NOTE 4229

Comparison of Internal and External Watchdog Timers

By: Eric Schlaepfer, Applications Engineer

Jun 05, 2008

Abstract: This article compares the benefits and drawbacks of internal (integrated with the microprocessor) and external (hardware-based) watchdog timers (WDTs). It explains that internal watchdog timers are easy to implement, but subject to failure. The MAXQ2000 microcontroller's WDT serves as an example of an internal watchdog. Hardware-based watchdog timers require additional board space, but are indispensable when reliability is paramount. This article concludes with a comparison matrix that highlights the main advantages and disadvantages of each WDT solution.

Introduction

A watchdog timer (WDT) forces an embedded microprocessor or microcontroller to reset (hardware reset) in response to an invalid software state. Such states can be as simple as a toggled register bit caused by a software bug, a cosmic ray, or an event such as EMI (electromagnetic interference).

This article examines some considerations for selecting the best timer for an application.

Typical Applications for a WDT

Preventing microprocessor lockup is a typical application for a WDT. Generally, embedded software has a "main loop" which calls into several subroutines that direct a variety of activities. The WDT is then reset once at the top of the loop. If the loop fails to execute properly for any reason, the watchdog will fire and the device will be reset.

A system with a WDT is particularly well-suited to detecting bit errors. Momentary bit errors are caused by interruptions, including soft memory failures and EMI discharges into memory devices and their interfaces. These bit errors cause temporary bit-polarity flipping of data into and out of the embedded processor. When this bit error occurs while fetching program information, the microprocessor will begin to execute erroneous code. Potentially, the processor could begin executing operands instead of op codes. When the processor begins executing this bad code, it will not properly execute the code that services the watchdog. After the timeout interval, the watchdog will cause a processor reset. In a properly designed system, this reset will correct the original error.

Also note that a WDT cannot detect a fault instantaneously. By definition, the WDT must reach the end of a predetermined time interval before it resets the processor. This fact explains why the shortest possible timeout interval should be selected to expire before the WDT can correct an out-of-control condition.

Internal vs. External WDTs

WDTs can be internal to a microprocessor, such as the MAXQ2000 microcontroller, external as a stand-alone IC, or embedded as part of a support ASIC. Whether the WDT is internal or external, each device type has advantages and disadvantages. The internal timer saves money, but can be affected by runaway code. While somewhat more costly to implement, an external WDT has a separate clock source that gives it better reliability; if properly configured, it cannot be bypassed or disabled by runaway code.

Basic WDT Operation

The heart of every WDT is a counter register. A clock source continuously increments the counter register, and when it overflows, the watchdog logic forces a system reset. To prevent this reset, you must periodically clear the counter register in a procedure called "servicing the watchdog." External WDT ICs have a digital input that clears the counter register on every low-to-high or high-to-low transition. For internal WDTs, the "servicing" implementations vary. The MAXQ2000's WDT, for example, can be serviced by setting a register bit (MOVE WDCN.0, #1). The PIC microcontroller and other microprocessors use a dedicated instruction (CLRWDT) to service the counter register.

Software operating correctly should provide digital signal transitions with sufficient regularity so that the counter register never overflows. But software can become stuck in an infinite loop or begin runaway execution so that the counter register is not cleared. That condition eventually triggers a reset. The WDT period is the time required for the counter to overflow. To prevent a reset, the intervals between the watchdog's servicing must be shorter than the timeout period.

Some WDTs include features more advanced than the basic counter-register architecture. A common variation is the "windowed" or "min/max" WDT. This feature signals a reset when the watchdog is serviced either too slowly or too quickly. If a signal to clear the counter register arrives while the number in the counter register is below some threshold, the device triggers a system reset. Simply put, the watchdog issues a system reset if it is serviced too rapidly, thereby preventing the microprocessor from operating if the crystal oscillator begins to run at a harmonic of the crystal's nominal frequency. Another feature disables the WDT through a logic input (external watchdog) or register bit (internal watchdog).

Watchdog Timers Internal to a Microprocessor

This short article cannot cover all variations of the microprocessor (µP) WDT, but instead examines the WDT internal to the MAXQ2000 microcontroller. While this timer is slightly more complex than the basic counter-register WDT described above, it is similar to WDTs incorporated on many other commercially available microcontroller chips.

If not serviced, the MAXQ2000's WDT overflows, triggers an interrupt, counts for 512 additional system-clock cycles, and then resets itself. That interrupt provides a "last chance" to save debugging information, a chance that most designers agree is useful during circuit development and troubleshooting. Instead of saving debugging information, moreover, the interrupt could be used to recover from the error and clear the watchdog. That latter approach, however, can compromise the system's reliability.

Like other internal WDTs, the MAXQ2000's watchdog can be disabled by software. Note, however, that this capability is a double-edged sword: runaway code can disable the watchdog and then continue its rampage.

Some microprocessors connect their WDT to an internal oscillator separate from the system clock. Although the clock for the MAXQ2000's WDT is derived from the system clock, the MAXQ2000's clock circuit will switch to a backup RC oscillator in the event of a failure in the main oscillator.

Watchdog Timer ICs

Dedicated, external WDT ICs offer a wide range of features and are available from many manufacturers. Some, like the MAX6752, have the "windowed watchdog" capability. The MAX6752 also lets you match the requirements of the microprocessor and application by using external capacitors to adjust the watchdog timeout period and the width of the reset pulse.

Logic input pins on the MAX6369 WDT can disable the timer and prevent resets. Although it is tempting to connect the logic pins to the microprocessor, doing so risks a compromise in system reliability because runaway code could accidentally disable the watchdog. Most often, the watchdog needs to be disabled during startup, while the microprocessor boots. The MAX6369 provides an extra-long watchdog period after coming out of reset. The normal-duration timeout then returns after the watchdog is serviced for the first time. This method is much more reliable than an explicit disabling and reenabling of the watchdog timer.

External watchdog timers can also monitor the V_{CC} supply, and assert a system reset when that voltage drops below a specified threshold. Some timers provide a debounced and ESD-protected reset-switch input. The voltage monitor in such ICs is generally more accurate over the operating temperature range than is the built-in brownout circuit included in some microprocessors.

Conclusion

Table 1 presents the benefits and drawbacks for external and internal WDTs. External watchdog-timer ICs are more expensive, and should, therefore, be used for critical systems that need higher reliability. However, connecting a WDT's disable input to a microprocessor (which allows the watchdog resets to be disabled by software), compromises the reliability of an internal watchdog timer. Thus, watchdog timers internal to microprocessors are suitable mainly for noncritical systems and for reducing costs.

Table 1. Watchdog-Timer Comparison Chart

	External WDTs	Internal WDTs
Pros	• If properly designed, cannot be disabled from software accidentally. • Clock source separate from system clock. • Available with min/max watchdog-timeout period. • Reset output can be connected to other system ICs that need a reset. • Monitors one or more power-supply rails (the internal brownout circuit found in some microprocessors is not very accurate). • Timeout period is fully adjustable by using capacitors.	• Cost is essentially zero. • To save debugging information, it can trigger an interrupt before resetting. • Convenient to use. • Sometimes a bidirectional reset pin can be connected to other system ICs that need reset. • Can modify timeout when the processor changes from high speed to low speed or sleep mode. • Watchdog timeout period can vary less with temperature.
Cons	• Requires some cost to implement. • Timeout period over temperature can vary substantially. • Requires one microprocessor I/O line. • Timeout must be calculated to match both high-speed and low-speed clock modes, due to microprocessor power-save feature.	• Almost all can be disabled by software (thus, not as reliable).

A similar article appeared in the September, 2007 edition of *Electronic Products Magazine*.

Related Parts		
MAX6752	µP Reset Circuits with Capacitor-Adjustable Reset/Watchdog Timeout Delay	Free Samples

More Information
For Technical Support: http://www.maximintegrated.com/support
For Samples: http://www.maximintegrated.com/samples
Other Questions and Comments: http://www.maximintegrated.com/contact

Application Note 4229: http://www.maximintegrated.com/an4229
APPLICATION NOTE 4229, AN4229, AN 4229, APP4229, Appnote4229, Appnote 4229