Detailed Design for the Watchdog Timer Driver

Overall Design

The watchdog timer has two modes of operation normal operation where the watchdog period is determined by the system processor clock or independent operation where there is a separate timer for the watchdog timer. The driver will configure the Renesas MCU according to the MCU interface specification.

Configuration

There is a register named OFS0 that is located in flash memory that is used to configure watchdog operations. The configurable parameters are:

1. Whether the watchdog is using the system clock for its timing or an external clock.
2. Whether the watchdog is restarts automatically after a reset or is just stopped after a reset.
3. The number of clock cycles in watchdog period.
4. A divide by counter for the clock being used for timing.
5. Window parameters for allowing a watchdog refresh to occur. The system can be configured to allow watchdog resets for a portion of the watchdog period in lieu of the entire watchdog period.
6. Whether the watchdog keeps counting in sleep mode.
7. Whether a reset is generated on a watchdog timeout or an interrupt is generated (non-maskable or regular).

There will be two configurations developed, one for normal product operation and one for software development. The distinction between the two is that the configuration for software development should allow for developers and other people involved with system development to have an opportunity to explore why a watchdog occurred when it happens and perhaps work around a problem temporarily.

Software Development Configuration

1. Watchdog is timed from the system clock. Whatever the timeout period is, we should be able to get close to it using the system clock. This will involve setting the number of clock cycles and divide by counter appropriately.
2. The watchdog will be stopped after a reset.
3. The window will be such that the watchdog can be refreshed at any time during the watchdog period.
4. The watchdog will not keep counting is sleep mode.
5. A non-maskable interrupt is generated when the watchdog times out.
6. Watchdog is disabled after a reset.

Product Operation Configuration

1. Watchdog is timed from the external clock. This will involve setting the number of clock cycles and divide by counter appropriately for the desired watchdog period.
2. The watchdog will be restart after a reset.
3. The window will be such that the watchdog can be refreshed at any time during the watchdog period.
4. The watchdog will keep counting is sleep mode.
5. A reset is generated when the watchdog times out.
6. Watchdog is enabled after a reset.

The watchdog timer driver will have three public methods:

1. Init() for initialization of the hardware-software interface.
2. KickWatchdog() for refreshing the watchdog timer before the timer expires
3. IsWatchdogReset() for determining whether or not the watchdog timed out causing a reset.

Init()

The watchdog timer is initialized as to its software development state or its normal production operation state. In normal product operation there is essentially nothing to do. Therefore a conditional compilation for the software development state is implied.

KickTheWatchdog()

Refresh the watch to prevent it from timing out.

IsWatchdogReset()

Returns true when a reset was caused by a watchdog timeout.