Order this document by EB314/D

Motorola Semiconductor Engineering Bulletin

EB314

Updating the Software Watchdog on M683xx and MC68HC16 Products

**By Charles Melear
Austin, Texas**

General Information

The software watchdog on MC683xx and MC68HC16 devices is used to detect improperly operating software.

The software watchdog is enabled from the release of external reset. That is, the SWE bit (bit 7 of the system protection control register at $xxFA21) is set to a logic 1 automatically.

The timeout periods of the software watchdog are selected by a combination of the SWP, SWT1, and SWT0 bits of the system protection control register.

If the software watchdog is not reset at appropriate intervals, it will time out and cause a software watchdog reset which is equivalent to a hardware reset. The software watchdog counter will be reset to its maximum value when the system software writes the value $55 followed by $AA to the software watchdog service register (SWSR) at $xxFA27.

If this sequence is not performed before the software watchdog decrements to $0000, a system reset will occur.

Some hardware considerations limit the rate at which software watchdog reset sequences can occur.

Specifically, when a write $55, write $aa sequence is written to the SWSR, the actual resetting of the software watchdog counter does not occur until the next falling edge of the software watchdog clock. Once the software watchdog counter is reset, further attempts to perform either a software watchdog reset sequence or to change the software watchdog timeout period are inhibited for the clock low time of the software watchdog clock period immediately following an actual reset of the watchdog counter.

Therefore, for proper and predictable operation of the software watchdog, do not update the SWSR at a rate which is more often than twice the period of the software watchdog counter clock source.

The software watchdog counter (see **Figure 1**) is driven by one of these clock sources:

- EXTAL divided by 128 in fast mode
- EXTAL in slow mode
- EXTAL divided by 128 when VCO is disabled

The clock period for the software watchdog timer, when using a 32.768-kHz crystal (slow mode) with the VCO/PLL enabled, is 30.5 microseconds.

The clock period of the software watchdog is also 30.5 microseconds when using a 4.194-MHz crystal in fast mode with the VCO/PLL enabled.

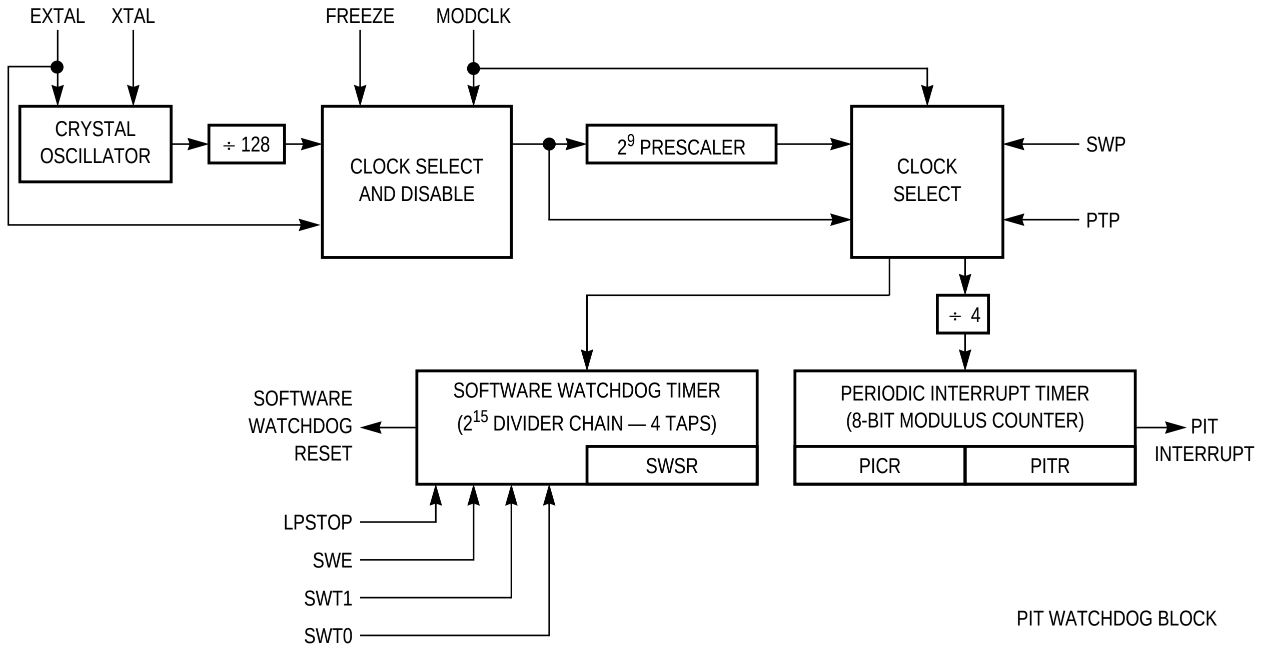

Figure 1. Periodic Interrupt Timer and Software Watchdog Timer

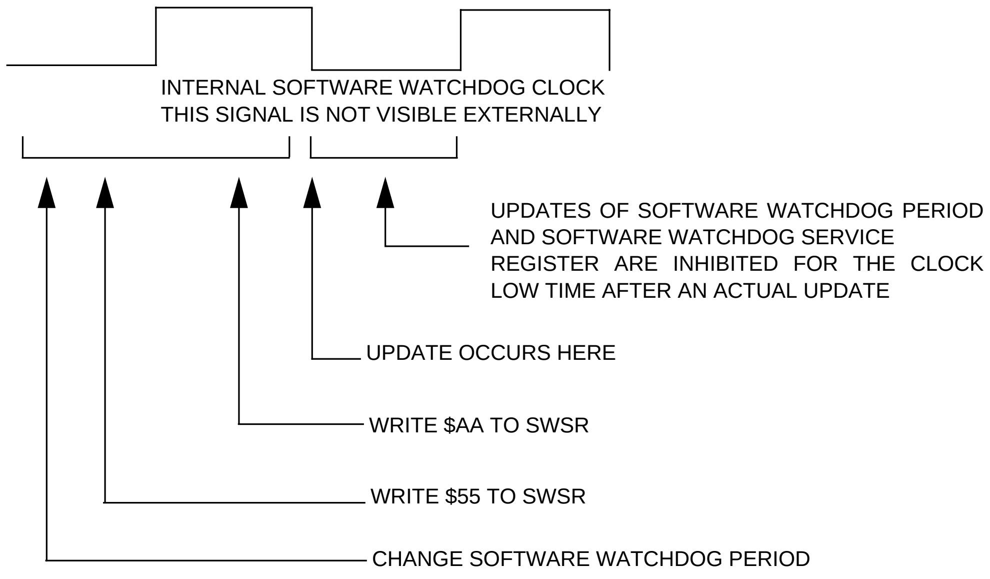
INTERNAL SOFTWARE WATCHDOG CLOCK
THIS SIGNAL IS NOT VISIBLE EXTERNALLY
UPDATES OF SOFTWARE WATCHDOG PERIOD AND SOFTWARE WATCHDOG SERVICE REGISTER ARE INHIBITED FOR THE CLOCK LOW TIME AFTER AN ACTUAL UPDATE
UPDATE OCCURS HERE
WRITE $AA TO SWSR
WRITE $55 TO SWSR
CHANGE SOFTWARE WATCHDOG PERIOD

Figure 2. Software Watchdog Clock Period

In the example shown in **Figure 2**, the software watchdog period is changed and the reset sequence for the software watchdog counter is performed during one cycle of the software watchdog clock.

The CPU has no time limit to perform the three accesses of the software watchdog registers other than these actions must take place before the watchdog counter decrements to $0000. The new watchdog period will take effect when the software watchdog counter is reset.

The sequence shown in **Figure 2** will produce the desired result of both resetting the watchdog counter and changing the watchdog period.

However, a similar example, shown in **Figure 3**, will produce erroneous results.

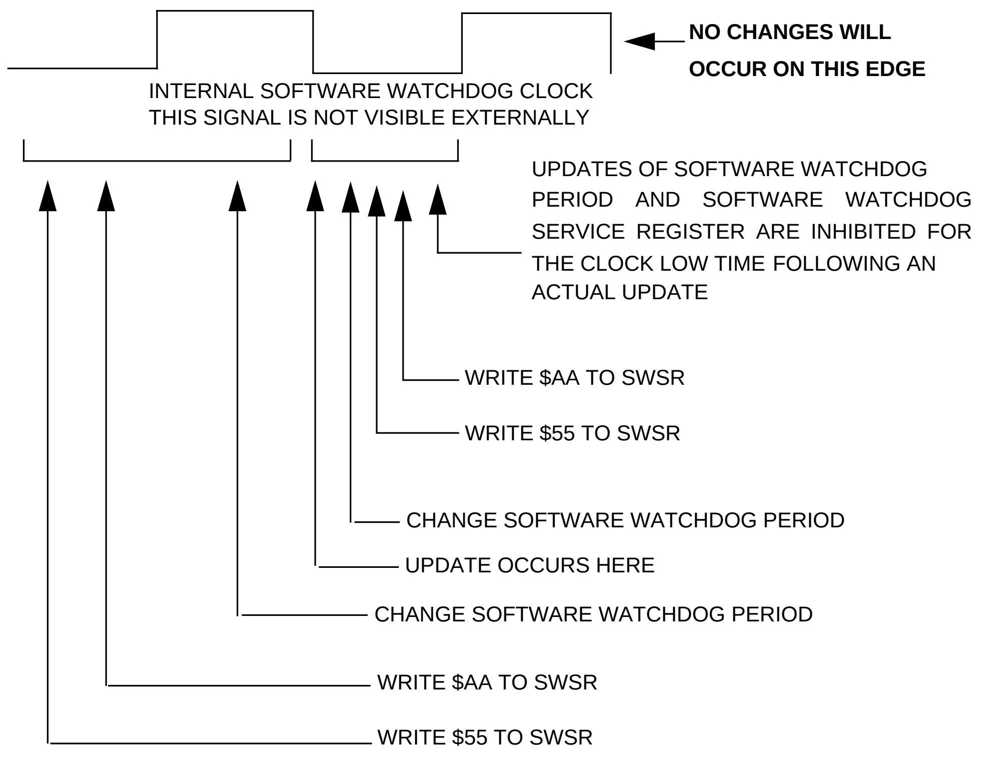
NO CHANGES WILL OCCUR ON THIS EDGE
INTERNAL SOFTWARE WATCHDOG CLOCK
THIS SIGNAL IS NOT VISIBLE EXTERNALLY
UPDATES OF SOFTWARE WATCHDOG PERIOD AND SOFTWARE WATCHDOG SERVICE REGISTER ARE INHIBITED FOR THE CLOCK LOW TIME FOLLOWING AN ACTUAL UPDATE
WRITE $AA TO SWSR
WRITE $55 TO SWSR
CHANGE SOFTWARE WATCHDOG PERIOD
UPDATE OCCURS HERE
CHANGE SOFTWARE WATCHDOG PERIOD
WRITE $AA TO SWSR
WRITE $55 TO SWSR

Figure 3. Software Watchdog Clock Period

In **Figure 3**, the software watchdog counter reset sequence is performed during one clock period of the software watchdog clock. The watchdog counter is properly reset on the next falling edge of the watchdog clock.

An attempt is then made to change the software watchdog period and also reset the watchdog counter in the clock low time after the update to the software watchdog service register.

The result is that the software watchdog will not be reset to its maximum count and the current timeout period will be used, not the timeout period that in this case is written during the clock low time immediately following an update. The new timeout period will be used the next time that a service of the SWSR is performed, providing that the watchdog counter does not decrement to $0000 first.

This particular feature can be problematic under certain conditions.

Consider the following program where the SWSR is updated during each iteration of the loop beginning on line 1. This creates a problem that prevents the software from properly changing the software watchdog period in line 6.

```
1 LOOP    move.w #$55,SWSR             ; first update of software service register
2         move.w #$AA,SWSR             ; second update of software service register
3         move syncr,D0
4         btst 3,D0                    ; test SLOCK bit to see if PLL is locked
5         bne LOOP
6         move.w #$new value,($sypcr)  ; change software watchdog period
7         move.w #$55,SWSR             ; first update of software service register
8         move.w #$AA,SWSR             ; second update of software service register
```

A reset of the watchdog counter occurs at line 2 during the first iteration of the loop. However, subsequent updates are ignored until after the low time of the next software watchdog clock has occurred. Then, the watchdog can be reset again. The loop created by lines 1 through 5 properly keeps the watchdog from decrementing to $0000 even though most of the accesses to the watchdog registers are ignored.

The software watchdog period is updated in line 6. However, the actual update (resetting the software watchdog counter to its maximum with the new value) does not occur until a `move #$55, move #$AA` sequence has occurred. Since there are only four instructions between the software watchdog update in line 2 and the update in line 8, the update in line 8 cannot occur.

For a successful update to occur, the instruction in line 2 would have to occur during one software watchdog clock period and the instruction in line 6 would have to occur after the software watchdog clock went high during the next software watchdog clock period.

If the system software and hardware depend upon a longer watchdog period due to the update in line 8, an anomalous situation is created. This happens because the update in line 8 will not occur because software watchdog updates are inhibited during the next software watchdog clock low time following an actual update. The shorter watchdog timeout period will be used but the software will expect a longer watchdog timeout period. This being the case, an unexpected software watchdog reset will occur.

This problem would not have occurred if the watchdog period had been changed prior to entering the loop formed by instructions 1 through 5.

Conclusion

Do not put a watchdog software service routine inside a loop and then attempt to change the software watchdog period and/or perform a reset sequence on the software watchdog service register immediately upon exiting the loop.

To ensure that changes to either the software watchdog period and/or reset sequences to the software watchdog service register occur correctly, a time interval of at least two clock periods of the internal software watchdog clock must elapse between each service of the software watchdog service register.

How to reach us:

USA/EUROPE/Locations Not Listed: Motorola Literature Distribution, P.O. Box 5405, Denver, Colorado 80217, 1-303-675-2140 or 1-800-441-2447. Customer Focus Center, 1-800-521-6274

JAPAN: Motorola Japan Ltd.; SPS, Technical Information Center, 3-20-1, Minami-Azabu, Minato-ku, Tokyo, 106-8573 Japan. 81-3-3440-8573

ASIA/PACIFIC: Motorola Semiconductors H.K. Ltd.; Silicon Harbour Centre, 2 Dai King Street, Tai Po Industrial Estate, Tai Po, N.T., Hong Kong. 852-26668334

Mfax™, Motorola Fax Back System: RMFAX0@email.sps.mot.com; http://sps.motorola.com/mfax/; TOUCHTONE, 1-602-244-6609; US and Canada ONLY, 1-800-774-1848

HOME PAGE: http://motorola.com/sps/

MOTOROLA