SECTION 11: PROGRAMMABLE TIMERS

All members of the High–Speed Microcontroller family incorporate three 16–bit programmable timers and some also have a Watchdog Timer with a programmable interval. Because the Watchdog Timer is significantly different from the other timers, it is described separately. The 16–bit Timers are referred to simply as timers.

In most modes, the timers can be used as either counters of external events or timers. When functioning as a counter, 1 to 0 transitions on a port pin are monitored and counted. When functioning as timers, they effectively count oscillator cycles. The time base for the timer function is the main oscillator clock divided by either 4 or 12. This selection is described below. Because each clock pulse must be sampled high for one machine cycle and low for one machine cycle to be recognized, this sets the maximum sampling frequency on any timer input at 1/8 of the main oscillator frequency.

The three timers are compatible with the 80C32. That is, they offer the same controls and I/O functions that were available in the 80C32. As mentioned above, the actual timing of these functions is user selectable to be compatible with the machine cycle of the older generation of 8051 family (12 clocks per tick) or the new generation (4 clocks per tick). The timing for each of the three timers can be selected independently and can be changed dynamically. Each timer has 4 primary modes as discussed below.

The Watchdog Timer Reset provides CPU monitoring by requiring software to clear the timer before the user selected interval expires. If the Timer is not cleared, the CPU will be reset by the Watchdog. The Watchdog function is optional and is described below. Since the High–Speed Microcontroller timers have a variety of features, the following summary table shows the capabilities.

Timer 0
13–bit Timer/counter
16–bit Timer/counter
8–bit Timer w/ Auto–reload
Two 8–bit Timer/counters
External control pulse timer/counter

Timer 1
13–bit Timer/counter
16–bit Timer/counter
8–bit Timer w/ Auto–reload
External control pulse timer/counter
Baud rate generator

Timer 2
16–bit Timer/counter
16–bit Timer with capture
16–bit Auto–reload Timer/counter
16–bit up/down auto–reload Timer/counter
Baud rate generator
Timer output clock generator

16–BIT TIMERS

Timers 0 and 1 are nearly identical. Timer 2 has several additional features such as up/down counting, capture values and an optional output pin that make it unique. Timers 0 and 1 are described first. Timer 2 is described separately. As mentioned above, the time base for each timer can be varied and this is also discussed in more detail below.

Timer 0 and 1 both have four operating modes. They are 13–bit timer/counter, 16–bit timer/counter, 8–bit timer/counter with auto–reload, and two 8–bit timers. The latter mode is available to Timer 0 only. These modes are controlled by the TMOD register. Each timer can also serve as a counter of external pulses (1 to 0 transition) on the corresponding Tn pin. This selection is controlled by the TMOD register. One other option is to gate the timer/counter using an external control signal. This allows the timer to measure the pulse width of external signals. Timers 0 and 1 are enabled using the TCON register which is also the location of their flags. The registers are described below. Following this is a detailed explanation of the four operating modes.

Each timer consists of a 16–bit register in two bytes. These are called TL0, TH0, TL1, and TH1. As shown, each timer is broken into low and high bytes. Software can read or write any of these locations at any time.

TMOD REGISTER SUMMARY

TIMER MODE CONTROL **TMOD; 89h**

TMOD.7 GATE – Timer 1 GATE control. When GATE=1, Timer 1 will clock only when $\overline{INT1}$ and TR1 =1. When GATE=0, Timer 1 will clock only when TR1=1 irrespective of $\overline{INT1}$.

TMOD.6 C/$\overline{T}$ – Counter/Timer select. When C/$\overline{T}$ is set to a 0, Timer 1 is incremented by internal clocks. When C/$\overline{T}$ is set to a 1, Timer 1 counts based on the T1 (P3.5) pin.

TMOD.5 M1 – Timer 1 Mode select bit 1.

TMOD.4 M0 – Timer 1 Mode select bit 0.

M1	M0	Mode
0	0	Mode 0 : 8–bits with 5–bit prescale
0	1	Mode 1 : 16–bits
1	0	Mode 2 : 8–bits with auto–reload
1	1	Mode 3 : Timer 1 stopped

TMOD.3 GATE – Timer 0 GATE control. When GATE=1, Timer 0 will clock only when $\overline{INT0}$ and TR0 =1. When GATE=0, Timer 0 will clock only when TR0=1 irrespective of $\overline{INT0}$.

TMOD.2 C/$\overline{T}$ – Counter/Timer select. When C/$\overline{T}$ is set to a 0, Timer 0 is incremented by internal clocks. When C/$\overline{T}$ is set to a 1, Timer 0 counts based on the T0 (P3.4) pin.

TMOD.1 M1 – Timer 0 Mode select bit 1.

TMOD.0 M0 – Timer 0 Mode select bit 0.

M1	M0	Mode
0	0	Mode 0 : 8–bits with 5–bit prescale
0	1	Mode 1 : 16–bits
1	0	Mode 2 : 8–bits with auto–reload
1	1	Mode 3 : Timer 0 is two 8–bit timers

TCON REGISTER SUMMARY

TIMER/COUNTER CONTROL	TCON; 88h
TCON.7	TF1 – Timer 1 overflow flag. Set to one when Timer 1 overflows from FFh, and cleared when the processor vectors to the interrupt service routine.
TCON.6	TR1 – Timer 1 run control. Turns on Timer 1 when this bit is set.
TCON.5	TF0 – Timer 0 overflow flag. Set to one when Timer 0 overflows from FFh, and cleared when the processor vectors to the interrupt service routine.
TCON.4	TR0 – Timer 0 run control. Turns on Timer 0 when this bit is set to a one.
TCON.3	IE1 – Interrupt 1 edge detect. Set by hardware when an edge/level is detected on $\overline{INT1}$.
TCON.2	IT1 – Interrupt 1 type select. $\overline{INT1}$ detects a falling edge when this bit is set to a 1. $\overline{INT1}$ detects a low level when this bit is a 0.
TCON.1	IE0 – Interrupt 0 edge detect. Set by hardware when an edge/level is detected on $\overline{INT0}$.
TCON.0	IT0 – Interrupt 0 type select. $\overline{INT0}$ detects a falling edge when this bit is set to a 1. $\overline{INT0}$ detects a low level when this bit is a 0.

MODE 0

Mode 0 configures either Timer 0 or Timer 1 for operation as a 13–bit Timer/Counter. As shown in figure 11–1, bits M1=0 and M0=0 of the TMOD register select this operating mode.

When using Timer 0, TL0 uses only bits 0–4. These bits serve as the 5 LSbs of the 13–bit timer. TH0 provides the 8 MSbs of the 13–bit timer. Bit 4 of TL0 is used as a ripple out to TH0 bit 0, thereby completely bypassing bits 5 through 7 of TL0. Once the timer is started using the TR0 (TCON.4) timer enable, the timer will count as long as GATE (TMOD.3) is 0 or GATE is 1 and pin $\overline{INT0}$ is 1. It will count oscillator cycles if C/$\overline{T}$ (TMOD.2) is set to a logic 0 and 1 to 0 transitions on T0 (P3.4) if C/$\overline{T}$ is set to a 1. When the 13–bit count reaches 1FFFh (all ones), the next count will cause it to roll over to 0000h. The TF0 (TCON.5) flag will be set and an interrupt will occur if enabled. The upper three bits of TL0 will be indeterminate.

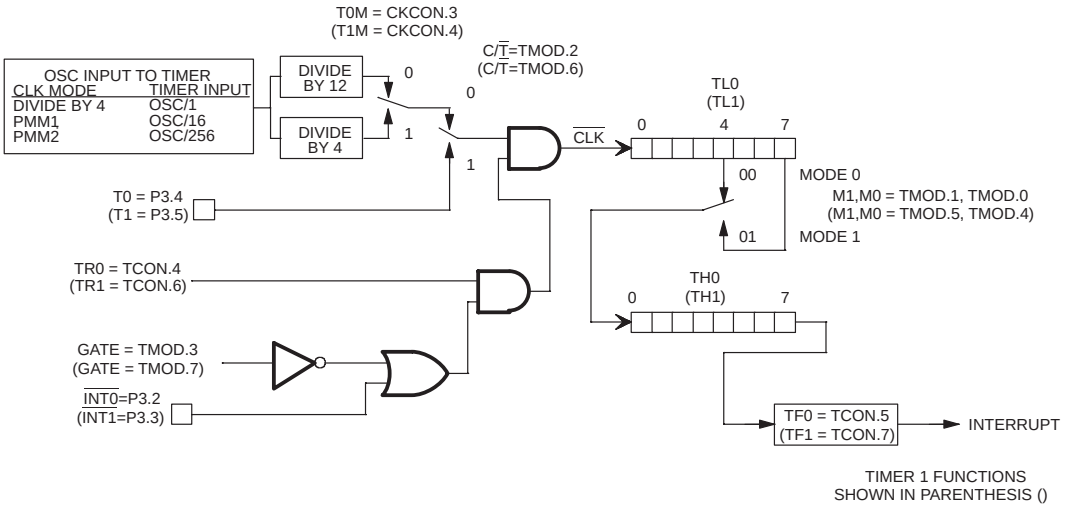
Note that when used as a timer, the time base may be either oscillator cycles/12 or oscillator cycles/4 as selected by bits TnM (n=0 or 1) of the CKCON register. This feature is described in more detail below.

Mode 0 operates identically when Timer 1 is used. The same information applies to TL1 and TH1, which form the 13–bit register. TR1 (TCON.6), $\overline{INT1}$ (P3.3), T1 (P3.5), and the relevant C/$\overline{T}$ (TMOD.6) and GATE (TMOD.7) bits have the same functions.

MODE 1

Mode 1 configures the timer for 16–bit operation as either a timer or counter. Figure 11–1 shows that bits M1=0 and M0=1 of the TMOD register select this operating mode. For Timer n, all of the TLn and THn registers are used. For example, if Timer 1 is configured in mode 1, then TL1 holds the LSB and TH1 holds the MSB. Rollover occurs when the timer reaches FFFFh. An interrupt will also occur if enabled and the relevant TFn flag is set. Timebase selection, counter/timer selection, and the gate function operate as described in mode 0.

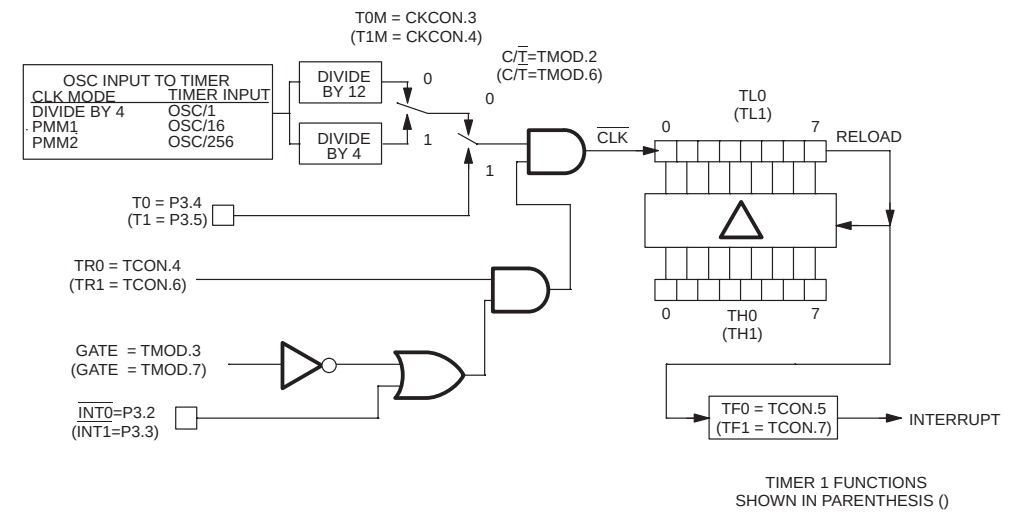
TIMER/COUNTER 0 AND 1 MODES 0 AND 1 Figure 11–1



MODE 2

This mode configures the timer as an 8–bit timer/counter with automatic reload of the start value. This configuration is shown in Figure 11–2, and is selected when bits M1and M0 of the TCON register are set to 1 and 0 respectively. When configured in Mode 2, the timer uses TLn to count and THn to store the reload value. Software must initialize both TLn and THn with the same starting value for the first count to be correct. Once the TLn reaches FFh, it will be automatically loaded with the value in THn. The THn value remains unchanged unless modified by software. Mode 2 is commonly used to generate baud rates since it runs without continued software intervention. As in modes 0 and 1, mode 2 allows counting of either oscillator cycles (crystal/12 or crystal/4) or pulses on pin Tn (C/$\overline{T}$=1) when counting is enabled by TRn and the proper setting of GATE and $\overline{INTn}$ pins.

TIMER/COUNTER 0 AND 1 MODE 2 Figure 11–2

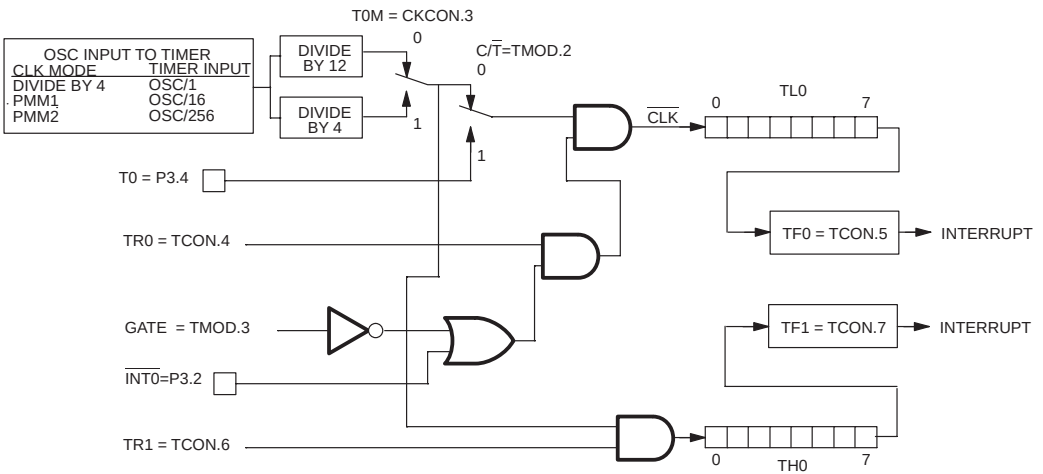


MODE 3

This mode provides an 8–bit timer/counter and a second 8–bit timer as indicated in Figure 11–3. In Mode 3, TL0 is an 8–bit timer/counter controlled by the normal Timer 0 bits (TR0=TCON.4 and TF0=TCON.5). TL0 can be used to count oscillator cycles (crystal/12 or crystal/4) or 1 to 0 transitions on pin T0 as determined by C/$\overline{T}$ (TMOD.2). As in the other modes, the GATE function can use $\overline{INT0}$ to give external run control of the timer to an outside signal.

TH0 becomes an independent 8–bit Timer in Mode 3, however it can only count oscillator cycles (divided by 12 or 4) as shown in the figure. In this mode, some of Timer 1's control signals are used to manipulate TH0. That is, TR1 (TCON.6) and TF1 (TCON.7) become the relevant control and flag bits associated with TH0.

TIMER/COUNTER 0 MODE 3 Figure 11–3



In Mode 3, Timer 1 stops counting and holds its value. Thus Timer 1 has no practical application while in Mode 3.

As mentioned above, when Timer 0 is in Mode 3, it uses some of Timer 1's resources (i.e., TR1 and TF1). Timer 1 can still be used in Modes 0, 1, and 2 in this situation, but its flexibility becomes somewhat limited. While it maintains its basic functionality, its inputs and outputs are no longer available. Therefore when Timer 0 is in Mode 3, Timer 1 can only count oscillator cycles, and it does not have an interrupt or flag. With these limitations, baud rate generation is its most practical application, but other time–base functions may be achieved by reading the registers.

TIMER 2

Like Timers 0 and 1, Timer 2 is a full function timer/counter, however it has several additional capabilities that make it more useful. Timer 2 has independent control registers in T2CON and T2MOD, and is based on count registers TL2 and TH2. All of these registers are described in detail below.

T2CON REGISTER SUMMARY

TIMER TWO CONTROL	**T2CON; C8h**
T2CON.7	TF2 – Timer 2 Overflow Flag. Hardware will set TF2 when the Timer 2 overflows from FFFFh or from the count equal to the capture register in down count mode. It must be cleared to 0 by software. TF2 will only be set to a 1 if RCLK and TCLK are both cleared to a 0.
T2CON.6	EXF2 – Timer 2 External Flag. Hardware will set EXF2 when a reload or capture is caused by a falling transition on the T2EX pin (P1.1). EXEN2 must be set for this function. This flag must be cleared to 0 by software. Writing a one to this bit will force a timer interrupt if enabled.
T2CON.5	RCLK – Receive Clock Flag. This bit determines whether Timer 1 or 2 is used for Serial Port 0 timing of received data in Serial Modes 1 or 3. RCLK =1 causes Timer 2 overflow to be used as the receive clock. RCLK=0 causes Timer 1 overflow to be used as the receive clock.
T2CON.4	TCLK – Transmit Clock Flag. This bit determines whether Timer 1 or 2 is used for Serial Port 0 timing of Transmit data in Serial Modes 1 or 3. TCLK=1 causes Timer 2 overflow to be used as the transmit clock. TCLK=0 causes Timer 1 overflow to be used as the transmit clock.
T2CON.3	EXEN2 – Timer 2 External Enable. Setting this bit to a 1 allows a capture or reload to occur as a result of a falling transition on T2EX (P1.1), if Timer 2 is not generating baud rates for the serial port. EXEN2=0 causes Timer 2 to ignore all external events at T2EX.
T2CON.2	TR2 – Timer 2 run. Setting this bit to a 1 starts Timer 2. Setting it to a 0 stops Timer 2.
T2CON.1	C/$\overline{T2}$ – Counter/Timer Select. Setting this bit to a 0 selects a timer function for Timer 2. Setting it to a 1 selects a counter of falling transitions on T2 (P1.0). Timer 2 runs at 4 clocks per tick or 12 clocks per tick as programmed by CKCON.5. This bit will be overridden and Timer 2 directed to use a divide by 2 clock if either the baud–rate generator or clock output mode is used.
T2CON.0	CP/$\overline{RL2}$ – Capture/Reload Flag. When this bit is set to 1, Timer 2 captures will occur on 1 to 0 transitions of T2EX (P1.1) if EXEN2=1. When this bit is set to 0, auto–reloads will occur when Timer 2 overflows or when 1 to 0 transitions occur on T2EX if EXEN2=1. If either RCLK or TCLK is set to a 1 this bit will not function and the timer will function in an auto–reload mode following each overflow.

T2MOD REGISTER SUMMARY

TIMER TWO MODE CONTROL	T2MOD; C9h
T2MOD.7–2	Reserved
T2MOD.1	T2OE – Timer 2 Output Enable. Setting this bit to a 1 enables the Timer 2 to drive the T2 (P1.0) pin with a clock output. When T2OE=0, the T2 (P1.0) pin is used as either an input for Timer 2 or a standard port pin.
T2MOD.0	DCEN – Down Count Enable. When this bit is set to 1, the Timer 2 function counts up or down when in 16–bit auto–reload mode depending on T2EX (P1.1). When DCEN is set to a 0, the Timer 2 counts up only.

TIMER 2 CAPTURE REGISTERS SUMMARY

LEAST SIGNIFICANT BYTE CAPTURE OF TIMER 2	RCAP2L; CAh
RCAP2L.7–0	This register is used to capture the TL2 value when Timer 2 is configured in capture mode. RCAP2L is also used as the LSB of a 16–bit reload value when Timer 2 is configured in auto–reload mode.

MOST SIGNIFICANT BYTE CAPTURE OF TIMER 2	RCAP2H; CBh
RCAP2H.7–0	This register is used to capture the TH2 value when Timer 2 is configured in capture mode. RCAP2H is also used as the MSB of a 16–bit reload value when Timer 2 is configured in auto–reload mode.

TIMER 2 MODES

As is seen in the register descriptions, Timer 2 has several abilities not found in Timers 0 and 1. However, it does not offer the 13–bit and dual 8–bit modes. Thus it runs in 16–bit mode at all times. Also note that instead of offering an 8–bit auto–reload mode, Timer 2 has a 16–bit auto–reload mode. This mode uses the Timer Capture registers to hold the reload values. The modes available on Timer 2 are described below.

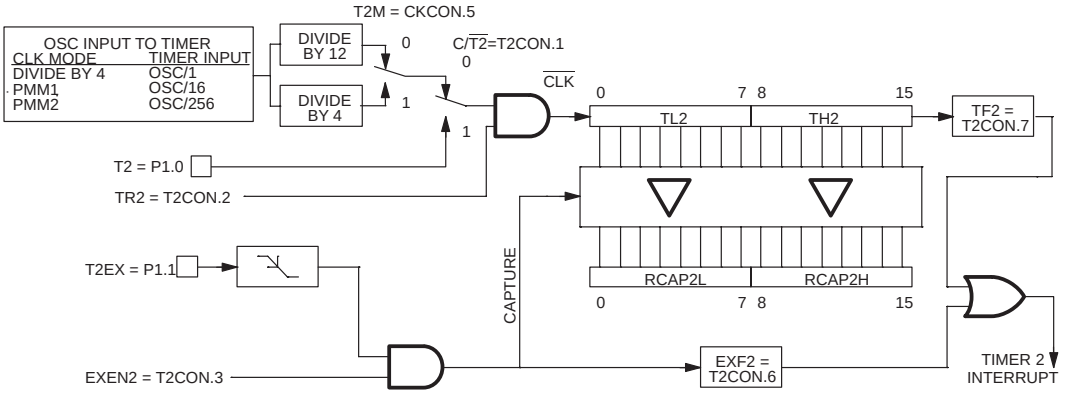
16–bit Timer/Counter

In this mode, Timer 2 performs a simple timer or counter function where it behaves similarly to mode 1 of Timers 0 and 1, but uses 16 instead of 8 bits. This mode, along with the optional capture mode described below, is illustrated in Figure 11–4. The 16–bit count values are found in TL2 and TH2 Special Function Registers (addresses 0CCh and 0CDh respectively). The selection of whether a Timer or Counter function is performed is made using the bit C/$\overline{T2}$ (T2CON.1). When C/$\overline{T2}$ is set to a logic 1, Timer 2 behaves as a counter where it counts 1 to 0 transitions at the T2 (P1.0) pin. When C/$\overline{T2}$ is set to a logic 0, Timer 2 functions as a timer where it counts the oscillator cycles divided by either 12 or 4 as determined by bit T2M (T2CON.5). Timing or counting is enabled by setting bit TR2 (T2CON.2) to 1, and disabled by setting it to 0. When the counter rolls over from FFFFh to 0000h, the TF2 flag (T2CON.7) is set and will cause an interrupt if Timer 2's interrupt is enabled.

16–bit Timer with Capture

A diagram of Timer 2's Capture Mode is shown in Figure 11–4. In this mode, the timer performs basically the same 16–bit timer/counter function described above. However, a 1 to 0 transition on T2EX (pin P1.1) causes the value in Timer 2 to be transferred into the capture registers if enabled by EXEN2 (T2CON.3). The capture registers, RCAP2L and RCAP2H, correspond to TL2 and TH2 respectively. The capture function is enabled by the CP/$\overline{RL2}$ (T2CON.0) bit. When set to a logic 1, the timer is in capture mode as described. When set to a logic 0, the timer is in auto–reload mode described later. As was possible with Timers 0 and 1, the timebase for Timer 2 may be selected to be oscillator cycles divided by either 12 or 4 when in this mode.

TIMER/COUNTER 2 WITH OPTIONAL CAPTURE Figure 11–4



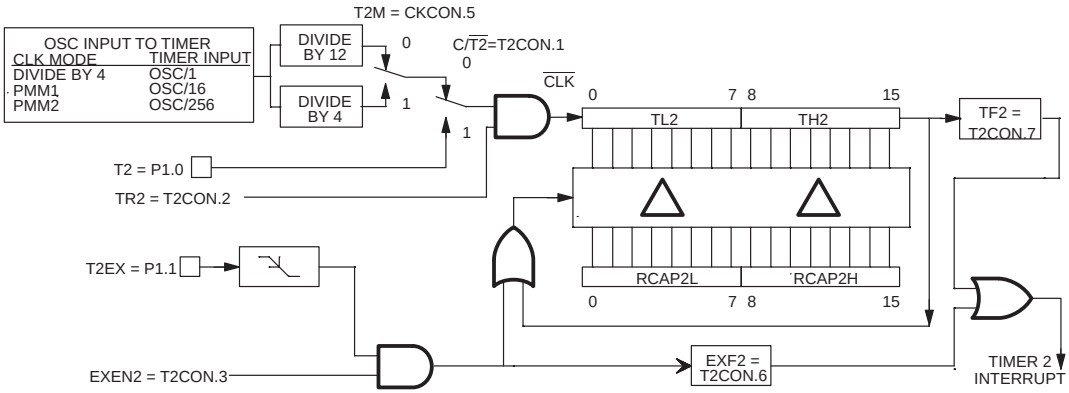
16–bit Auto–reload Timer/Counter

This mode is illustrated in Figure 11–5a. When Timer 2 reaches an overflow state, i.e., rolls over from FFFFh to 0000, it will set the TF2 Flag. This flag can generate an interrupt if enabled. In addition, the timer will restore its starting value and begin timing (or counting) again. The starting value is preloaded by software into the capture registers RCAP2L and RCAP2H. These registers cannot be used for capture functions while also performing auto–reload, so these modes are mutually exclusive. Auto–reload is invoked by the CP/$\overline{RL2}$ (T2CON.0) bit. When set to a logic 0, the timer is in auto–reload mode. When CP/$\overline{RL2}$ is set to a logic 1, the timer is in capture mode described above. If the oscillator timebase is used (C/$\overline{T2}$=T2CON.1=0), the timer's input may be selected to be oscillator cycles divided by either 12 or 4 as determined by T2M (CKCON.5). Otherwise, pulses on pin T2 (P1.0) are counted when C/$\overline{T2}$=1. As in other modes, Counting or timing is enabled or disabled with TR2 (T2CON.2).

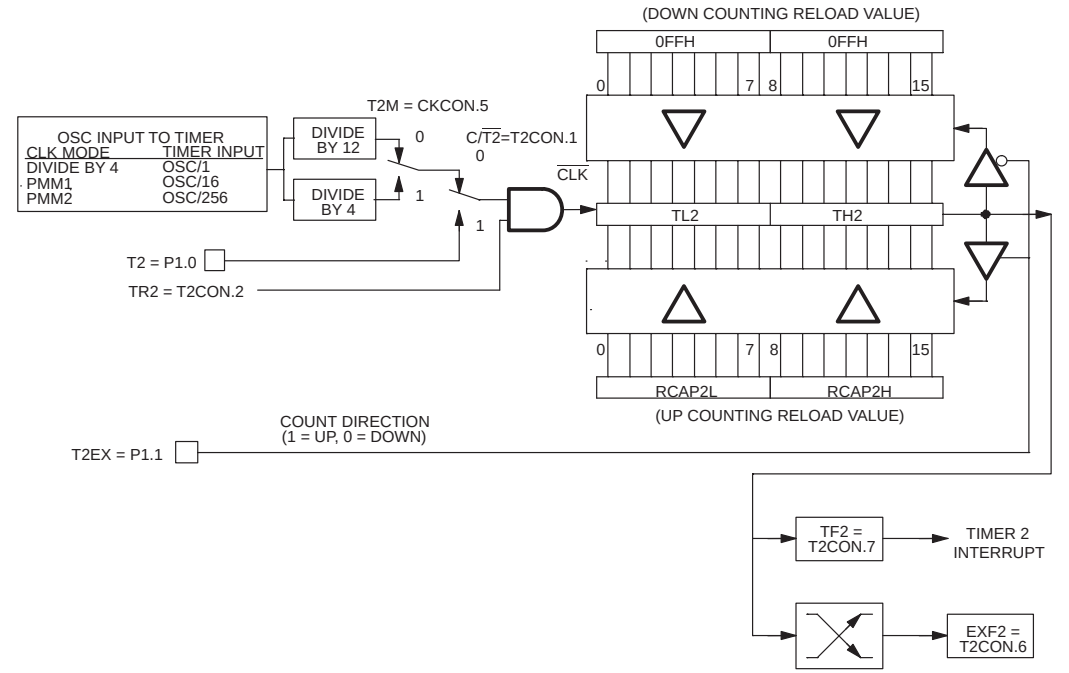
When in auto–reload mode, Timer 2 can also be forced to reload with the T2EX (P1.1) pin. A 1 to 0 transition will force a reload if enabled by the EXEN2 (T2CON.3) bit. If EXEN2 is set to a logic 1, then a 1 to 0 transition on T2EX will cause a reload. Otherwise, the T2EX pin will be ignored.

TIMER/COUNTER 2 AUTO RELOAD MODE Figure 11–5

(a) DCEN=0



(a) DCEN=1



Up/Down Count Auto–reload Timer/Counter

The up/down auto–reload counter option is selected by the DCEN (T2MOD.0) bit, and is illustrated in Figure 11–5b. When DCEN is set to a logic 1, Timer 2 will count up or down as controlled by the state of pin T2EX (P1.1). T2EX will cause upward counting when a logic 1 is applied and down counting when a logic 0 is applied. When DCEN=0, Timer 2 only counts up.

When an upward counting overflow occurs, the value in RCAP2L and RCAP2H will load into T2L and T2H. In the down count direction, an underflow occurs when T2L and T2H match the values in RCAP2L and RCAP2H respectively. When an underflow occurs, FFFFh is loaded into T2L and T2H and counting continues.

Note that in this mode, the overflow/underflow output of the timer is provided to an edge detection circuit as well as to the TF2 bit (T2CON.7). This edge detection circuit toggles the EXF2 bit (T2CON.6) on every overflow or underflow. Therefore, the EXF2 bit behaves as a seventeenth bit of the counter, and may be used as such.

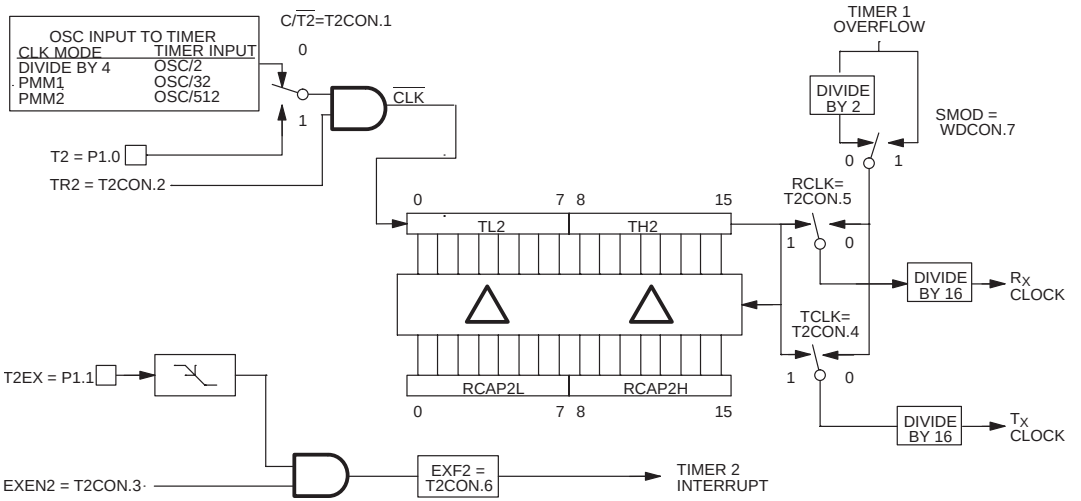
Baud Rate Generator

Timer 2 can be used to generate baud rates for Serial Port 0 in serial modes 1 or 3. Baud rate generator mode is invoked by setting either the RCLK or TCLK bit in the T2CON register to a logic 1, as illustrated in Figure 11–6. In this mode, the timer continues to function in auto–reload mode, but instead of setting the interrupt flag T2F (T2CON.7) and potentially causing an interrupt, the overflow generates the shift clock for the serial port function. As in normal auto–reload mode, an overflow causes RCAP2L and RCAP2H to be transferred into T2L and T2H respectively. Note that when RCLK or TCLK is set to 1, the Timer 2 is forced into 16–bit auto–reload mode regardless of the CP/$\overline{RL2}$ bit.

As explained above, the timer itself cannot set the T2F interrupt flag and therefore cannot generate an interrupt. However if EXEN2 (T2CON.3) is set to 1, a 1 to 0 transition on the T2EX (P1.1) pin will cause the EXF2 (T2CON.6) interrupt flag to be set. If enabled, this will cause a Timer 2 Interrupt to occur. Therefore in this mode, the T2EX pin may be used as an additional external interrupt if desired.

Another feature of the baud rate generator mode is that the crystal derived timebase for the timer is the crystal frequency divided by 2. No other crystal divider selection is possible. If a different timebase is desired, bit C/$\overline{T2}$ (T2CON.1) may be set to a 1 sourcing the timebase from an external clock source supplied by the user on pin T2 (P1.0). Software should not access TL2 or TH2 while the timer is running (TR2=1) in baud rate generator mode. In this mode the timer is clocking so fast that a software read of or write to the TL2 and TH2 registers may corrupt the timer. The RCAP registers may be read, but not modified, while TR2=1. Stop the timer (TR2=0) to modify these registers.

TIMER/COUNTER 2 BAUD RATE GENERATOR MODE Figure 11–6



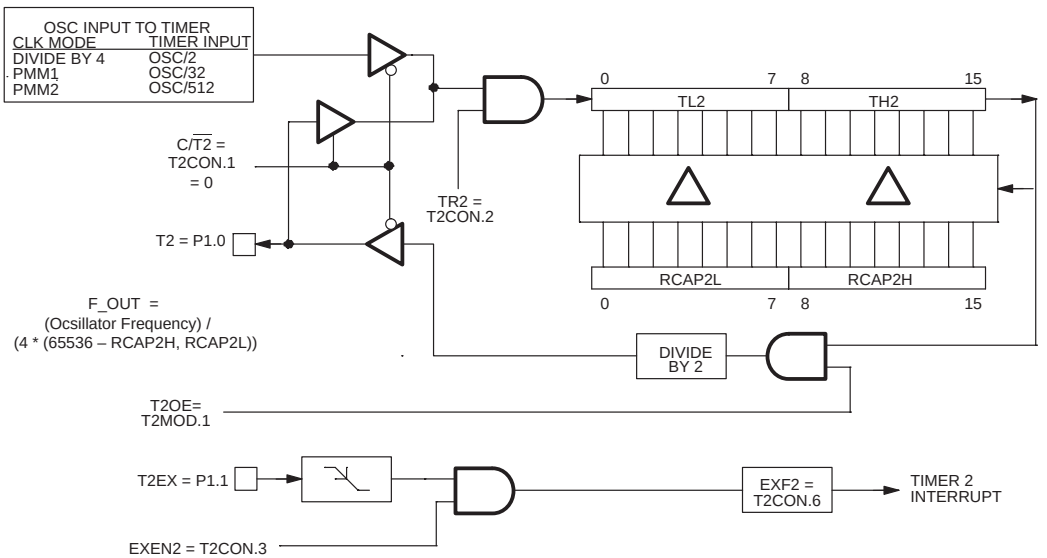
Timer Output Clock Generator

Timer 2 can also be configured to drive a clock output on port pin P1.0 (T2) as shown in Figure 11–7. To configure Timer 2 for this mode, first it must be set to 16–bit auto–reload timer mode (CP/$\overline{RL2}$=0, C/$\overline{T2}$=0). Next, the T2OE (T2MOD.1) bit must be set to a logic 1. TR2 (T2CON.2) must also be set to a logic 1 to enable the timer.

This mode will produce a 50% duty cycle square wave output. The frequency of the square wave is given by the formula in the figure. Each timer overflow causes an edge transition on the pin, i.e., the state of the pin toggles.

Note that this mode has two somewhat unique features in common with the baud rate generation mode. First, the timebase is the crystal frequency divided by 2, and no other divider selection is possible. Second, the timer itself will not generate an interrupt, but if needed, an additional external interrupt may be caused using T2EX as described above. Because of the two mode's similarities, the timer can be used to generate both an external clock and a baud rate clock simultaneously. Once the clock out mode is established, either TCLK or RCLK is set to 1, and the RCAP2 registers are loaded, the timer will provide a clock to both functions.

TIMER/COUNTER 2 CLOCK OUT MODE Figure 11–7



TIME–BASE SELECTION

The High–Speed Microcontroller allows the user to select either 4 or 12 clocks as the time–base for each timer independently. When using the 16–bit Timer/Counters in timer mode, the timer/counter counts the oscillator cycles divided by a predetermined number. In the standard 8051, the 8051 timers count the oscillator divided by 12, which is the standard 8051 machine cycle timing. The High–Speed Microcontroller allows the option of setting the timers to operate from a divide by 4 of the input clock to allow higher precision timing and faster baud rates. This selection has no effect on CPU timing, only on the timers. Following a reset, the timers default to 12 clocks as the time–base to remain drop–in compatible with the original 8051.

The 4 or 12 clock decision is independent for each timer and the default is 12 clocks per timer tick. As an example, a user might select both the baud rate generator timer and one other timer to run at 12 clocks per timer tick with the third timer at 4 clocks per tick. This allows one timer to measure higher speed events or to gain better resolution. The control bits for the time–base selection are located in the Clock Control register (CKCON;8Eh). Timer 2 will function at 2 clocks per tick

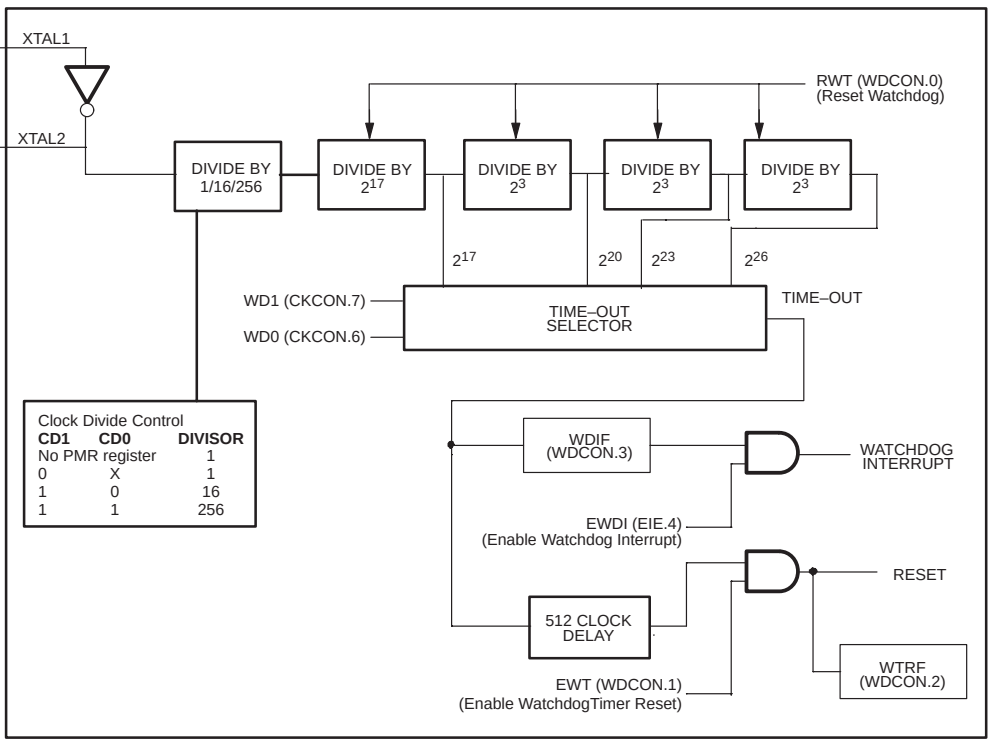
when set for baud rate generation or clock output as described above.

The use of Power Management modes will affect the input clock to the timer as shown in the illustrations. In general, they will divide the input clock by either 16 or 256 for PMM1 and PMM2, respectively. Timer 2, when operating in Baud Rate Generator or Clock Out mode normally uses the input clock frequency divided by 2, but when PMM1 and PMM2 are used, it will operate from a time–base of the input clock divided by 32 and 512, respectively.

WATCHDOG TIMER

The Watchdog Timer is a user programmable clock counter that can serve as a time–base generator, an event timer, or a system supervisor. As can be seen in the diagram of Figure 11–8, the timer is driven by the main system clock that is supplied to a series of dividers. The divider output is selectable, and determines the interval between time–outs. When the time–out is reached, an interrupt flag will be set, and if enabled, a reset will occur. The interrupt flag will cause an interrupt to occur if its individual enable bit is set and the global interrupt enable is set. The reset and interrupt are completely discrete functions that may be acknowledged or ignored, together or separately for various applications.

WATCHDOG TIMER Figure 11–8



The Watchdog Timer Reset function works as follows. After initializing the correct time–out interval (discussed below), software first restarts the Watchdog using RWT (WDCON.0) and then enables the reset mode by setting the Enable Watchdog Timer Reset (EWT=WDCON.1) bit. At any time prior to reaching its user selected terminal value, software can set the Reset Watchdog Timer (RWT=WDCON.0) bit. If RWT is set before the time–out is reached, the timer will start over. If the time–out is reached without RWT being set, the Watchdog will reset

the CPU. Hardware will automatically clear RWT after software sets it. When the reset occurs, the Watchdog Timer Reset Flag (WTRF=WDCON.2) will automatically be set to indicate the cause of the reset, however software must clear this bit manually.

The Watchdog Timer is a free running timer. When used as a simple timer with both the reset and interrupt functions disabled (EWT=0 and EWDI=0), the timer will continue to set the Watchdog Interrupt flag each time the timer completes the selected timer interval as programmed by WD1 (CKCON.7) and WD0 (CKCON.6). Restarting the timer using the RWT (WDCON.0) bit, allows software to use the timer in a polled time–out mode. The WDIF bit is cleared by software or any reset.

The Watchdog Interrupt is also available for applications that do not need a true Watchdog Reset but simply a very long timer. The interrupt is enabled using the Enable Watchdog Timer Interrupt (EWDI=EIE.4) bit. When the time–out occurs, the Watchdog Timer will set the WDIF bit (WDCON.3), and an interrupt will occur if the global interrupt enable (EA=IE.7) is set. Note that WDIF is set 512 clocks before a potential Watchdog Reset. The Watchdog Interrupt Flag will indicate the source of the interrupt, and must be cleared by software.

Using the Watchdog Interrupt during software development can allow the user to select ideal watchdog reset locations. Code is first developed without enabling the Watchdog Interrupt or Reset functions. Once the program is complete, the Watchdog Interrupt function is enabled to identify the required locations in code to set the RWT (WDCON.0) bit. Incrementally adding instructions to reset the Watchdog Timer prior to each address location (identified by the Watchdog Interrupt) will allow the code to eventually run without receiving a Watchdog Interrupt. At this point the Watchdog Timer Reset can be enabled without the potential of generating unwanted resets. At the same time the Watchdog Interrupt may also be disabled. Proper use of the Watchdog Interrupt with the Watchdog Reset allows interrupt software to survey the system for errant conditions.

When using the Watchdog Timer as a system monitor, the Watchdog Reset function should be used. If the Interrupt function were used, the purpose of the watchdog would be defeated. For example, assume the system is executing errant code prior to the Watchdog Interrupt. The interrupt would temporarily force the system back into control by vectoring the CPU to the interrupt service routine. Restarting the Watchdog and exiting by an RETI or RET, would return the processor to the lost position prior to the interrupt. By using the Watchdog Reset function, the processor is restarted from the beginning of the program, and therefore placed into a known state.

The Watchdog has four time–out selections based on the input crystal frequency as shown in the figure. The selections are a preselected number of clocks. Therefore, the actual time–out interval is dependent on the crystal frequency. Shown below are the four time–outs with some example periods for different crystal speeds. Note that the time period shown is for the interrupt event. The reset, when enabled, will occur 512 clocks later regardless of whether the interrupt is used. Therefore the actual Watchdog time–out period is the number shown below plus 512 clocks. Watchdog generated resets will last for two machine cycles.

WD1	WD0	WATCHDOG INTERVAL	NUMBER OF CLOCKS	TIME AT 1.8432 MHz	TIME AT 11.0592 MHz	TIME AT 16 MHz	TIME AT 20 MHz	TIME AT 25 MHz
0	0	2^{17}	131072	71.11 ms	11.85 ms	8.19 ms	6.55 ms	5.24 ms
0	1	2^{20}	1048576	568.89 ms	94.81 ms	65.54 ms	52.43 ms	41.94 ms
1	0	2^{23}	8388608	4551.11 ms	758.52 ms	524.29 ms	419.43 ms	335.54 ms
1	1	2^{26}	67108864	36408.88 ms	6068.15 ms	4194.30 ms	3355.44 ms	2684.35 ms

The Watchdog time–out selection is made using bits WD1 (CKCON.7) and WD0 (CKCON.6) as shown in the figure. The time–out selections possible are shown in the bit descriptions that follow. The watchdog timeout period is affected by the use of Power Management modes. The slower clock rate, either divide by 64 or divide by 1024 is used as the input source for the watchdog timer. This allows the watchdog period to remain synchronized with device operation.

As discussed above, the Watchdog Timer has several SFR bits that contribute to its operation. It can be enabled to function as either a reset source, interrupt source, software polled timer or any combination of the three. Both the reset and interrupt have status flags. The Watchdog also has a bit that restarts the timer. A summary table showing the bit locations is below. A description follows.

BIT NAME	DESCRIPTION	REGISTER LOCATION	BIT POSITION
EWT	Enable Watchdog Timer Reset	WDCON – D8h	WDCON.1
RWT	Reset Watchdog Timer	WDCON – D8h	WDCON.0
WD1	Watchdog interval 1	CKCON – 8Eh	CKCON.7
WD0	Watchdog interval 0	CKCON – 8Eh	CKCON.6
WTRF	Watchdog Timer Reset Flag	WDCON – D8h	WDCON.2
EWDI	Enable Watchdog Timer Interrupt	EIE – E8h	EIE.4
WDIF	Watchdog Interrupt Flag	WDCON – D8h	WDCON.3

The Watchdog Timer is a free running timer. It will be disabled by a Power–fail Reset. A Watchdog time–out reset will not disable the Watchdog Timer, but will restart the timer. In general, software should set the Watchdog to whichever state is desired, just to be certain of its state. Control bits that support Watchdog operation are described below.

WDCON REGISTER SUMMARY

WATCHDOG CONTROL **WDCON; D8h**

WDCON.3 WDIF – Watchdog Interrupt Flag. If the Watchdog Interrupt is enabled (EIE.4), hardware will set this bit to indicate that the Watchdog Interrupt has occurred. If the interrupt is not enabled, this bit indicates that the time–out has passed. If the Watchdog Reset is enabled (WDCON.1), the user has 512 clocks to strobe the Watchdog prior to a reset. Software or any reset can clear this flag.

WDCON.2 WTRF – Watchdog Timer Reset Flag. Hardware will set this bit when the Watchdog Timer causes a reset. Software can read it, but must clear it manually. A Power–fail Reset will also clear the bit. This bit assists software in determining the cause of a reset. If EWT=0, the Watchdog Timer will have no affect on this bit.

WDCON.1 EWT – Enable Watchdog Timer Reset. Setting this bit will turn on the Watchdog Timer Reset function. The interrupt will not occur unless the EWDI bit in the EIE register is set. A reset will occur according to the WD1 and WD0 bits in the CKCON register. Setting this bit to a 0 will disable the reset but leave the timer running.

WDCON.0 RWT – Reset Watchdog Timer. This bit serves as the strobe for the Watchdog function. During the time–out period, software must set the RWT bit if the Watchdog is enabled. Failing to set the RWT will cause a reset when the

time–out has elapsed. There is no need to set the RWT bit to a 0 because it is self–clearing.

Read/write access: All bits have unrestricted read access. POR, EWT, WDIF, and RWT require a Timed Access write. The remaining bits have unrestricted write access.

CLOCK CONTROL **CKCON; 8Eh**

CKCON.7 WD1 – Watchdog Timer mode select bit 1. See table below for operation.

CKCON.6 WD0 – Watchdog Timer mode select bit 0. See table below for operation.

The WD select bits determine the time–out period of the Watchdog Timer. The timer divides the crystal frequency by a programmable value as shown below. The divider value is expressed in number of clock (crystal) cycles. Note that the reset time–out is 512 clocks longer than the interrupt, regardless of whether the interrupt is enabled.

WD1	WD0	Interrupt Divider	Reset Divider
0	0	2^{17}	$2^{17} + 512$
0	1	2^{20}	$2^{20} + 512$
1	0	2^{23}	$2^{23} + 512$
1	1	2^{26}	$2^{26} + 512$

The default Watchdog time–out is the shortest one (WD1=WD0=0). Software can change this value easily, so this should cause no inconvenience. However, the EWT, WDIF, and RWT bits are protected under the Timed Access procedure. This prevents software from accidentally enabling or disabling the Watchdog. Most importantly, it prevents errant code from accidentally clearing and restarting the Watchdog. More details are discussed in the section on Timed Access.