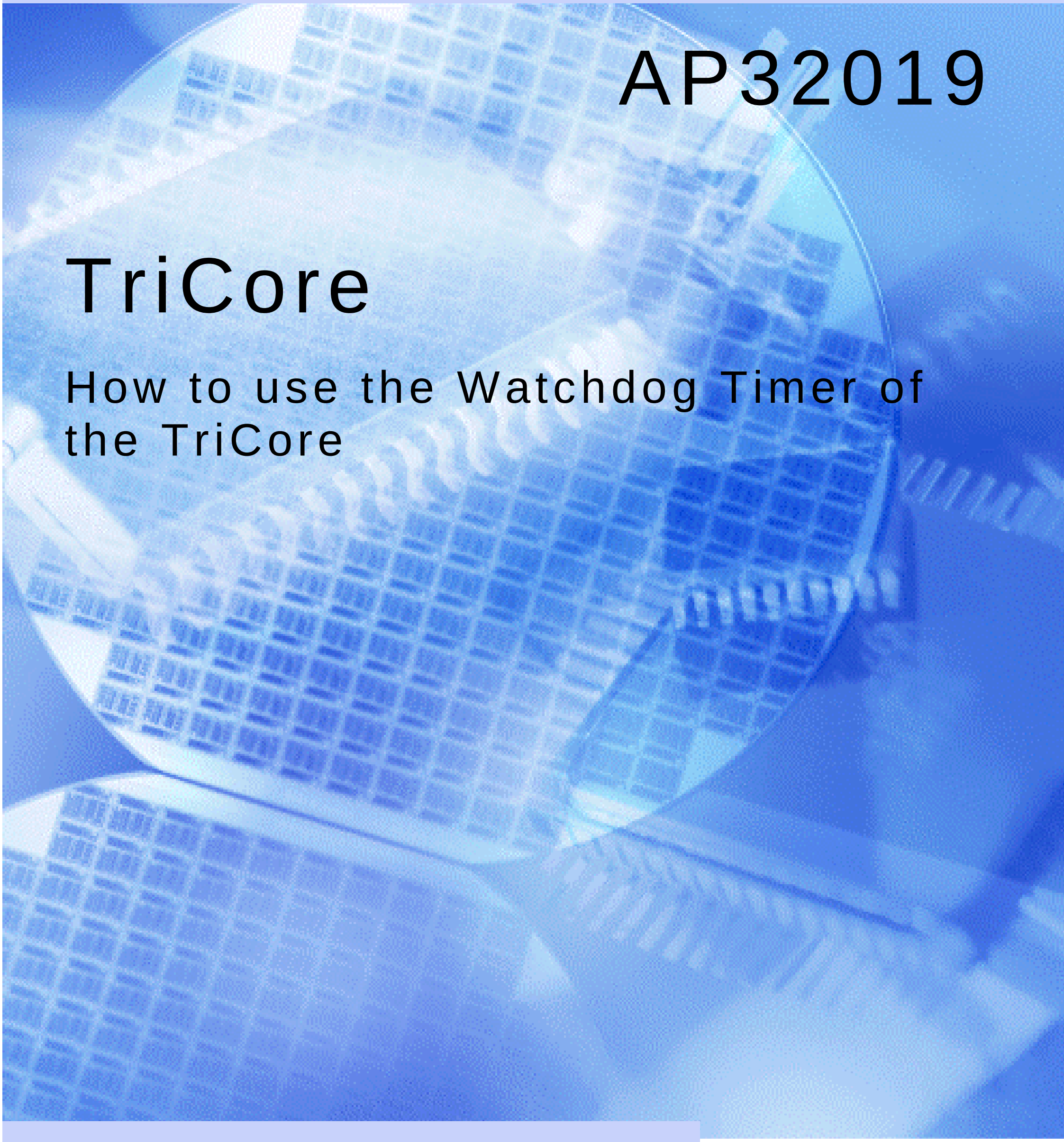
Application Note, V 2.2, Sept. 2002

AP32019

TriCore

How to use the Watchdog Timer of the TriCore



Microcontrollers

Never stop thinking.

Revision History:	**Sept. 2002**	**V 2.2**
Previous Version:	Jan. 2000	V 2.0

Page	Subjects (major changes since last revision)
all	Changed to new Infineon template
all	Logo changed to Infineon

We Listen to Your Comments

Any information within this document that you feel is wrong, unclear or missing at all? Your feedback will help us to continuously improve the quality of this document. Please send your proposal (including a reference to this document) to:
mcdocu.comments@infineon.com

Edition 2002-09

Published by
Infineon Technologies AG
81726 München, Germany

Table of Contents **Page**

1 Overview

Infineon TriCore 32-bit microcontrollers provide a Watchdog Timer (WDT) on-chip. This application gives hints and examples for in-system programming of the WDT and describes the WDT protocol.

The TriCore Watchdog Timer provides a recovery mechanism from software or hardware failure. The Watchdog Timer can either be used in an application, with its highly reliable and secure Watchdog function, or it can be disabled.

If the software fails to service this timer correctly, the TriCore is reset.

When the software is configured to always service the Watchdog Timer before it overflows, the Watchdog Timer will Time-out if the program does not progress properly. The Watchdog Timer also times out if the software error was due to hardware-related failures. This operation prevents the controller from malfunctioning for longer than a user-specified time.

Service of the Watchdog Timer is a critical system function, because unintentional service due to an error condition could disable the Watchdog function. Thus, the TriCore's Watchdog service requires a special instruction sequence with a password mechanism in order to become effective. Incorrect sequences also lead to a reset of the chip.

In order for software to determine the cause of a Watchdog failure, first the Watchdog Timer generates a NMI trap, then the reset occurs after a certain Time-out Period. In the NMI trap routine, critical system state such as the PC value, the stack pointer, and context pointers can be stored to data memory for examination after the reset has occurred. The Watchdog Timer also has flags that indicate the type of Watchdog failure.

In addition to the Watchdog function, a Watchdog Timer control register contains the WDTCON0.ENDINIT bit, which controls access to system critical registers. The state of the ENDINIT bit can be modified; however, this must be performed using the sophisticated password scheme of the Watchdog Timer. The Watchdog Timer monitors modifications of the ENDINIT bit such that after clearing ENDINIT, the Watchdog Timer enters a defined Time-out Mode; ENDINIT must be set again before the Time-out expires.

In systems where the Watchdog function is not applicable, the Watchdog Timer can be disabled. However, this operation is also protected via the password scheme. The Watchdog Time-out Function invoked on modifications of the ENDINIT bit is independent of the disable or enable status of the Watchdog Timer.

2 WDT Initialization

After a reset, the WDTCON0.ENDINIT bit is cleared, giving access to critical system registers protected via the ENDINIT bit. WDTCON1, used to control the operation of the Watchdog Timer, is one of the registers protected by ENDINIT.

When changing the operation of the Watchdog Timer through the controls in register WDTCON1, these changes have no immediate effect. They go into effect only after the ENDINIT bit has been set to 1 again. In this way, changes of these bits do not interfere with the Time-out Operation of the WDT.

Because the WDT is in the Time-out Mode after reset, there is a time limit of 65536 system clocks on the initialization of the critical system registers. The ENDINIT bit must be set to 1 before the Time-out expires.

1.1 Default Values

After a reset, the contents of the Watchdog Timer registers are as shown in Table 1.

Table 1 Watchdog Timer Default Values After Reset

Register	Default Contents	Description
WDTCON0	[31:16] 1111 1111 1111 1100 [15:0] 0000 0000 0000 0010	Reload value is 0xFFFC, WDTPW is 0; WDTCON0 is locked (WDTLCK=1); ENDINIT is 0, so access to ENDINIT-protected registers is enabled.
WDTCON1	[31:16] 0000 0000 0000 0000 [15:0] 0000 0000 0000 0000	Watchdog Timer disable request is 0; input clock request set to fSYSCLK/16384.
WDTSR	[31:16] 1111 1111 1111 1100 [15:0] 0000 0000 0001 00xx	The Watchdog counter contains 0xFFFC (the initial Time-out value); WDT is operating in Time-out Mode (WDTTO=1); WDT is enabled (WDTDS=0); input clock is fSYSCLK/16384. Bits WDTOE and WDTAE are set to 0 after a power-on hardware reset or software reset. In case of a reset caused by the WDT, these two bits are set depending on the error condition that caused the Watchdog reset.

2.1 Time Out Mode and Reset Pre-warning Mode

Time-out Mode is automatically triggered by the following events:

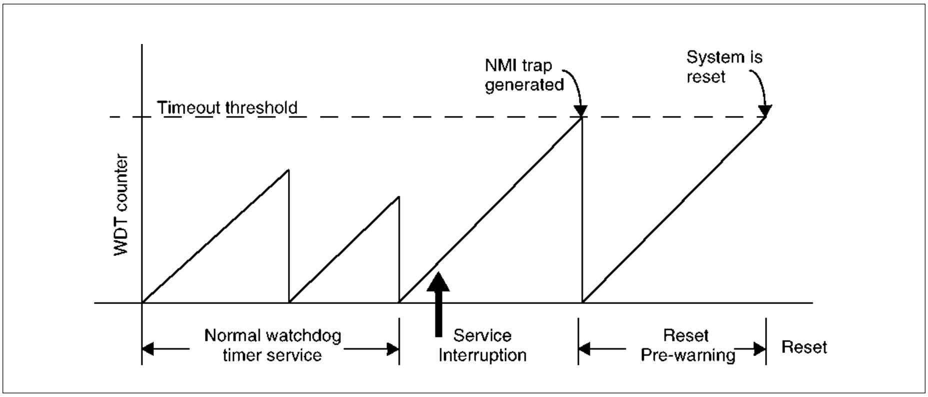
- Reset
- WDTCON0 is unlocked by a successful password access

Time-out Mode is indicated by WDTSR.WDTTO=1. As the name suggests, Time-out Mode lasts for a particular duration, called the Time-out Period. Time-out Mode must be exited before this period is over, or the system is reset.

To disable the Time-out Mode, bit ENDINIT in register WDTCON0 must be written to 1 with a modifying access. This access stops the Time-out, resets bit WDTSR.WDTTO, and switches the WDT back to the operation determined through WDTSR.WDTDS, WDTSR.WDTIS and WDTCON0.WDTREL. If WDTDS=1, the Watchdog Timer is reset to 0 and is stopped. If WDTDS=0, the Watchdog Timer is set to the value of WDTCON0.WDTREL and starts counting upwards with the clock input selected through WDTIS.

If the Time-out Mode is not properly stopped, the WDT overflows from 0xFFFF to 0x0000 and enters the Reset Pre-warning Mode. This mode, indicated through bit WDTSR.WDTPR=1, is similar to the Time-out Mode except that it is no longer possible to stop the Time-Out Period.

Note: Even when the WDT is disabled and ENDINIT is not set to 1 the Watchdog Timer is automatically set to Time-out Mode.



Reset Pre-warning Mode is automatically triggered by the following events:

- Watchdog Overflow Error (WDTSR.WDTOE=1)
- Watchdog Access Error (WDTSR.WDTAE=1)

In this mode a NMI trap request to the CPU is activated. The NMISR.NMIWDT bit in the NMI status register is set. When the Watchdog Timer again overflows from 0xFFFF to 0x0000, a reset will occur.

In this mode, a NMI trap routine can save critical system state such as the PC value, the stack pointer, and context pointers can be stored to data memory for examination after the reset has occurred. The Watchdog Timer also has flags that indicate the type of Watchdog failure. This function is especially important during program debugging.

In the NMI routine, a check of the bit NMISR.NMIWDT should always be performed to distinguish the NMI trap cause from an external NMI request.

Note: NMIWDT has to be reset by software.

2.2 Servicing the Watchdog Timer

If the Watchdog Timer is used in an application and is enabled (WDTSR.WDTDS=0), it must be regularly serviced to prevent it from overflowing.

Service is performed in two steps. First, the proper password must be written to WDTCON0 to unlock it. Password access to WDTCON0 automatically switches the WDT to Time-out Mode. Thus, the modifying access must be performed before the Time-out expires or a system reset will result. During the following modifying access, the strict requirement is that WDTCON0.ENDINIT as well as bit 1 and bits 7:4 are written with 1's, while bits 3:2 are written with 0's, as shown in the chapter Modify Access.

Changes to the reload value WDTCON0.WDTREL, or the user-definable password WDTCON0.WDTPW, are not required. However, changing WDTPW is recommended so that software can monitor Watchdog Timer service operations throughout the duration of an application program. If WDT service is properly executed, Time-out Mode is terminated, and the Watchdog Timer switches back to its former mode of operation, and WDT service is complete.

3 The access to the Watchdog Timer Registers

WDTSR is a read-only register and can be read at any time. Write accesses to this register have no effect (no error is reported in such a case). Updating of the status bits in register WDTSR is handled automatically in hardware. The two error flags, WDTCON1.WDTOE and WDTCON1.WDTAE, are not cleared through a Watchdog reset (but through any other reset). They are cleared with a successful access to WDTCON0 which sets ENDINIT to 1.

WDTCON1 can also be read at any time without any restrictions. Writing to this register is only possible if bit ENDINIT is cleared to 0 (WDTCON1 is ENDINIT-protected). Updates made to the bits in this register will go into effect after ENDINIT has been set to 1 again.

Access to WDTCON0 can be for any combination of the following reasons:

- To perform routine service to forestall a Watchdog Timer overflow
- To change fields in WDTCON0
- To change fields in WDTCON1
- To change other ENDINIT-protected system registers

All of these require unlocking WDTCON0 first. Proper access to WDTCON0 always requires two write accesses. The first write, called the password access, unlocks WDTCON0. The second write, called the modifying access, can change values in WDTCON0. When the modifying access completes, WDTCON0 is locked again automatically. If the modifying access sets WDTCON0.ENDINIT to 0, then other protected system registers, such as WDTCON1, are unlocked and can be modified.

3.1 Password Access

The password required to unlock WDTCON0 is comprised of the state of bits in WDTCON0 and WDTCON1, as indicated in Table 2:

Table 2 Password Requirements

Password Bit	Required Value
0	State of ENDINIT in register WDTCON0
1	Inverted state of WDTLCK in register WDTCON0
2	State of WDTIR in register WDTCON1
3	State of WDTDR in register WDTCON1
7:4	Always 1
15:8	Value of WDTPW in register WDTCON0
31:16	Value of WDTREL in register WDTCON0

determined just by reading the contents of the Watchdog registers. A modifying step that alters some of the bits is required in order to get the right password.

This password has to be written to the address location of WDTCON0 in order to unlock this register for modifications. If the password matches the requirements, register WDTCON0 is unlocked after this write operation has finished. This unlocked condition is indicated by WDTCON0.WDTLCK=0.

Now, the following write access can modify register WDTCON0. After this write has finished, WDTCON0 is automatically locked again. WDTLCK is set to 1 by hardware.

If the password value written to WDTCON0 during the password access does not match the contents of WDTCON0, a Watchdog access error situation exists. The Watchdog error situations are detailed in the section entitled “Watchdog Error Conditions" further below.

The following program in C-Code is an example how to unlock the register WDTCON0. It can be included as a procedure in the mainfunction:

```
//Password Access to unlock WDTCON0 include this in your project
void WDT_Passwd(void)
{
  #define WDTCON0 *((volatile unsigned int*) 0xF0000020)
  #define WDTCON1 *((volatile unsigned int*) 0xF0000024)

  unsigned int passwd;
  passwd = WDTCON0;                    //load value from WDTCON0
  passwd = passwd | 0x0f0;             //sets bit 7:4 to 1
  passwd = (WDTCON1 & 0x00c)           //bit 2:3 from WDTCON1
         | (passwd & 0xfffffff3);      //and bit 2:3 cleared from WDTCON0
  passwd = (passwd ^ 0x00000002);      //invert bit 1
  WDTCON0 = passwd;                    //write password to WDTCON0
}

//Procedure call
WDT_Passwd();
```

After this password sequence the WDTCON0 will be unlocked and you can change it by a modify access.

3.2 Modify Access

During the modifying access, the only strict requirement is that bits 1 and 7:4 are written to 1’s, and bits 3:2 are written to 0’s. All other bits can be set to user-definable values.

If the value written to WDTCON0 during the modifying access does not match these requirements, a Watchdog access error exists. The Watchdog error situations are detailed in the section entitled “Watchdog Error Conditions” further below.

The following program in C-Code is an example how to change the register WDTCON0. It can be included as a procedure in the mainfunction:

```
//Modify Access to WDTCON0 include this in your project
void WDT_Modify(unsigned long modify, unsigned long mask)
{
  #define WDTCON0 *((volatile unsigned int*) 0xF0000020)
  unsigned int newvalue;      //by the mask selected bits will be
                              //changed by the variable modify
  newvalue = WDTCON0 & (~mask));
  newvalue = newvalue | (modify & mask);
  newvalue = ((0xf2 | newvalue) & (0xfffffff3));//bit 1 and 7:4 are
                       //written to 1’s and bit 3:2 written to 0’s
  WDTCON0 = newvalue;  //write newvalue to WDTCON0
}
```

Now you can call the above mentioned procedure with the following instructions:

```
Mask = 0x0000000f;     //bits that will be changed set to 1
Modify = 0x0000000f;   //new value for WDTCON0, ENDINIT will be set to
WDT_Modify(Modify, Mask);//Procedure call
```

Note: *The values of the variables “Mask” and “Modify” can be changed. This is only an example and is described in the section entitled “How to change the WDTCON0”.*

3.3 Watchdog Error Conditions

There are two classes of Watchdog Timer Errors:

- **Watchdog Access Error:** an incorrect access to WDTCON0 was attempted
- **Watchdog Timer Overflow Error:** the WDT counter overflowed

The Watchdog Access Error Status Flag, WDTSR.WDTAE, is set in case of an access error. A Watchdog Access Error is generated when:

- An illegal password is supplied during password access to register WDTCON0
- Improper guard bits are supplied during a modifying access to WDTCON0

The Watchdog Overflow Error Status Flag, WDTSR.WDTOE, is set if an overflow error occurs. A Watchdog Timer Overflow Error is generated if the WDT counter overflows. In all error cases, the Watchdog Timer:

- Generates an NMI trap request
- Enters the Reset Pre-warning Mode

- Sets the Watchdog Reset Pre-warning Flag, WDTSR.WDTPR and Watchdog Time-out Period Indication Flag, WDTSR.WDTTO

Note: When reset is caused by the WDT, the two error flags, WDTSR.WDTOE and WDTSR.WDTAE, are not cleared. This allows the WDT to detect the Double Watchdog Error condition. Any other reset cause will clear these error flags. They are also cleared when WDTCON0.ENDINIT is set to 1.

4 How to change the Watchdog Timer Registers

4.1 How to change the WDTCON0

The first step to change the WDTCON0 register is to unlock it with the Password Access by a procedure as stated further above. This unlocked condition is indicated by WDTLCK=0. The register WDTCON0 is now unlocked and you can proceed with a modify access.

It is recommended to use the procedure `WDT_Modify(Mask, Modify)` further above, in order to avoid protocol errors. The following tables will show you how to set the values of the variables “Mask” and “Modify”. The result values for the variables “Mask” and “Modify” must be combined by a disjunction of the individual values.

Note: You can configure only the WDTCON registers.

Table 3 How to set the variable Mask

Which value do you want to change?	Required Value for variable “MASK”
ENDINIT	0x0000 000F
Reload Value	0xFFFF 0000
User-definable Password	0x0000 FF00
Example Result value for the variable “Mask” to change the value of ENDINIT and the Reload Value (combined by a disjunction of the individual values)	0xFFFF 000F

Table 4 How to set the variable Modify

Value of the variable	Value	Required Value for variable “Modify”
ENDINIT	0 1	0x0000 0000 0x0000 000F
Reload Value YYYY is a user-definable hexvalue of 16bit from 0x0000 to 0xFFFF (see chapter “Normal Timer Period” for hints)	YYYY	0xYYYY 0000

User-definable Password ZZ is a User-definable Hexvalue of 8bit from 0x00 to 0xFF (see chapter “Extended Security” for hints how to set this value)	ZZ	0x0000 ZZ00
Example Result value for the variable “Modify” to change the value of the ENDINIT bit to 1 and the Reload Value to 0xABCD (combined by a disjunction of the individual values)		0xABCD 000F

Now you can use the procedure `WDT_Modify(Mask, Modify)` with the calculated values. The register WDTCON0 will be changed and locked after this access again.

Example:

```
Mask = 0xffff000f;             //bits that will be changed set to 1
Modify = 0xabcd000f;           //new value for WDCON0
                               //(here ENDINIT will be set to 1
                               //and Reset Value to 0xabcd)
WDT_Modify(Modify, Mask);      //Procedure call
```

4.2 How to change the WDTCON1

Writing to the register WDTCON1 is only possible if bit ENDINIT is cleared (WDTCON1 is ENDINIT-protected). You can change only the Watchdog Timer Input Frequency Request Control Bit (WDTIR) and the Watchdog Timer Disable Request Control Bit (WDTDR).

Table 5

Field	Bits	Type	Description
WDTDR	3	rw	Watchdog Timer Disable Request Control Bit. 0 Request to enable the Watchdog Timer. 1 Request to disable the Watchdog Timer. As long as ENDINIT is 0, bit WDTDS in register WDTSR controls the current enable/disable status of the Watchdog Timer. When ENDINIT is 1, WDTDS is updated with the state of bit WDTDR.
WDTIR	2	rw	Watchdog Timer Input Frequency Request Control Bit. 0 Request to set input frequency to f_{SYSCLK} /16384. 1 Request to set input frequency to f_{SYSCLK} /256. As long as ENDINIT is 0, bit WDTIS in register WDTSR controls the current input frequency of the Watchdog Timer. When ENDINIT is 1, WDTIS is updated with the state of bit WDTIR.

Note: Updates made to the bits in this register will go into effect after WDTCON0.ENDINIT has been set to 1 again.

4.3 Normal Timer Period

The duration of Normal Timer Mode can be varied by two parameters: the input clock and the reload value.

The system clock, f_{SYSCLK}, can be divided by either 256 or 16384. WDTSR.WDTIS selects the input clock divider. The default value of WDTIS after reset is 0, corresponding to a frequency of f_{SYSCLK} /16384. When the Watchdog Timer is serviced in Normal Timer Mode, it is reloaded with the 16-bit reload value, WDTCON0.WDTREL.

The Watchdog Timer period can be varied over a wide range.

The maximum time period is achieved by setting WDTREL=0x0000. The minimum time period is achieved by setting WDTREL=0xFFFF. The general form with variable reload value WDTREL for these calculations is:

$$\text{Period} = (2^{16} - \text{WDTREL}) * 256 * 2^{(1 - \text{WDTIS}) * 6} / f_{SYSCLOCK}$$

Example:

WDTREL = 0xFFFC = 65532; WDTIS = 1; $f_{SYSCLOCK}$ = 100MHz;

Period = $(2^{16} – \text{WDTREL}) * 256 * 2^{(1 - \text{WDTIS}) * 6} / f_{SYSCLOCK}$

Period = $(2^{16} – 65532) * 256 * 2^{(1 - 1) * 6} / 100\text{MHz}) = 10{,}24\ \mu s$

5 Using the Watchdog Timer

Normally, an embedded application program is built around a loop of routines, which are repeatedly executed. For example, consider the configuration in the figure shown below. This illustrates a program which, after reset, carries out whatever initialization is required. Then it proceeds to a loop of routines that are executed over and over again. Besides this main loop, there is a set of interrupt service routines (these are not shown in the following figures).

Within the loop of the mainline program, the Watchdog Timer service sequences are embedded. Of course, also an initialization of the Watchdog must be made if a different time-out value than the default one is desired.

The WDT Service should be placed properly in the application such that a safe service before the next overflow is guaranteed. The service should normally be performed in the main routine. Placing the service sequence into an interrupt routine could cause that the WDT Service will be executed even if the main program goes awry. This could disable the intended monitoring function of the Watchdog Timer.

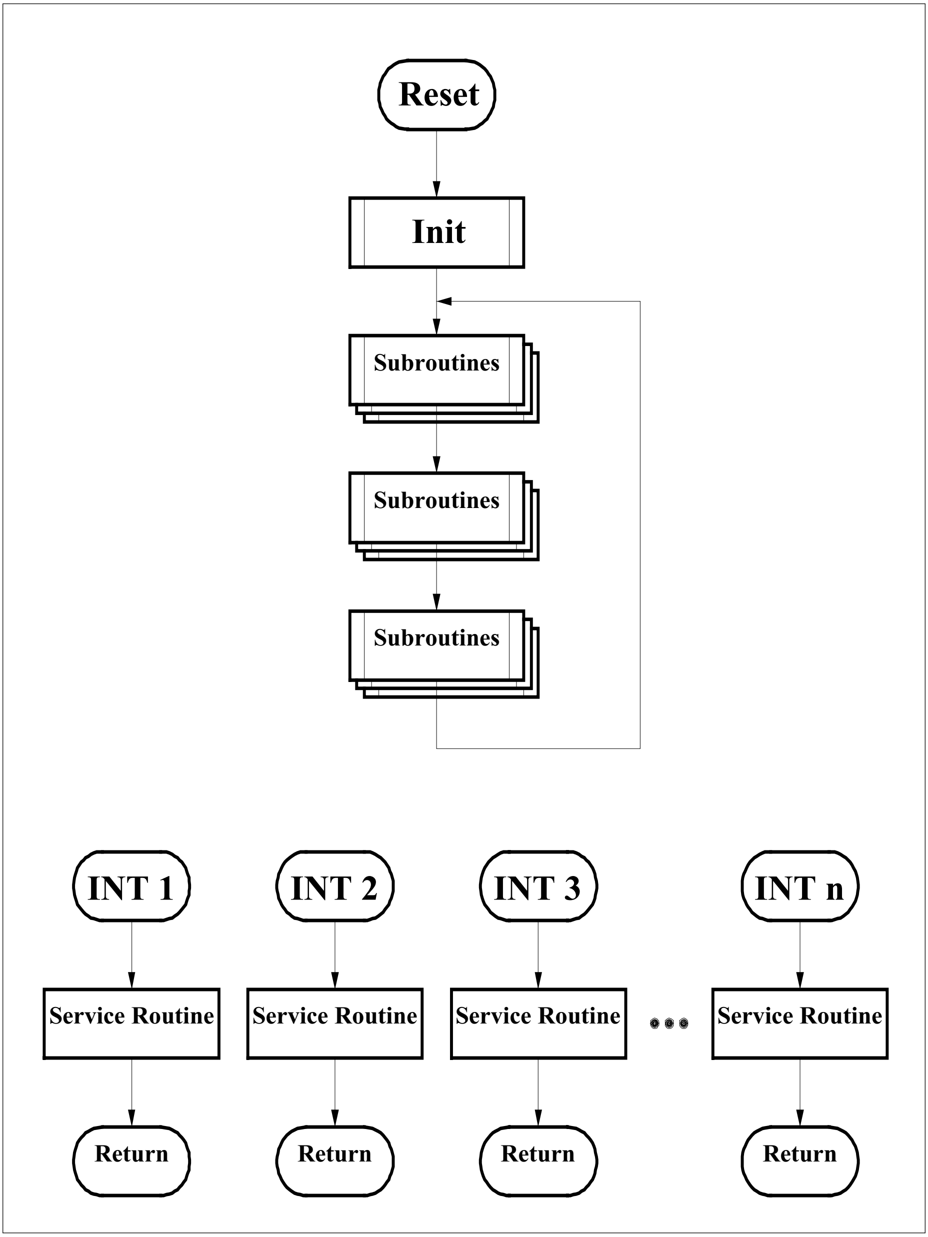


Figure 1 Standard WDT Service

6 Extended Security Thoughts

To increase the security of the WDT functionality, there are different possibilities how to handle these Watchdog services, described in the following sections. In the associated figures, only the update of the password field WDTPW in register WDTCON0 is shown.

6.1 Multiple Service Sequences

One option is to place several service sequences inside the main code. In order to monitor the program flow, one changes the WDT password within each of the service sequences, as shown in Figure 2. Since at the beginning of each sequence, first the *old password* has to be written to WDTCON0, a link to the previous service is performed. If due to an error, one or more of the service sequences are skipped, WDTCON0 will not contain the expected value at the beginning of the next service sequence: An Watchdog Timer Access Error will occur and the bit WDTAE is set. This will also happen if a service sequence is executed twice due to an erroneous loop. Of course, with this scheme, the service sequences must be placed at points in the code, which are *always* executed during the main loop. If you want to use this feature in your application you have to use the procedure `WDT_Pass_Secure(Oldpassword)` further below.

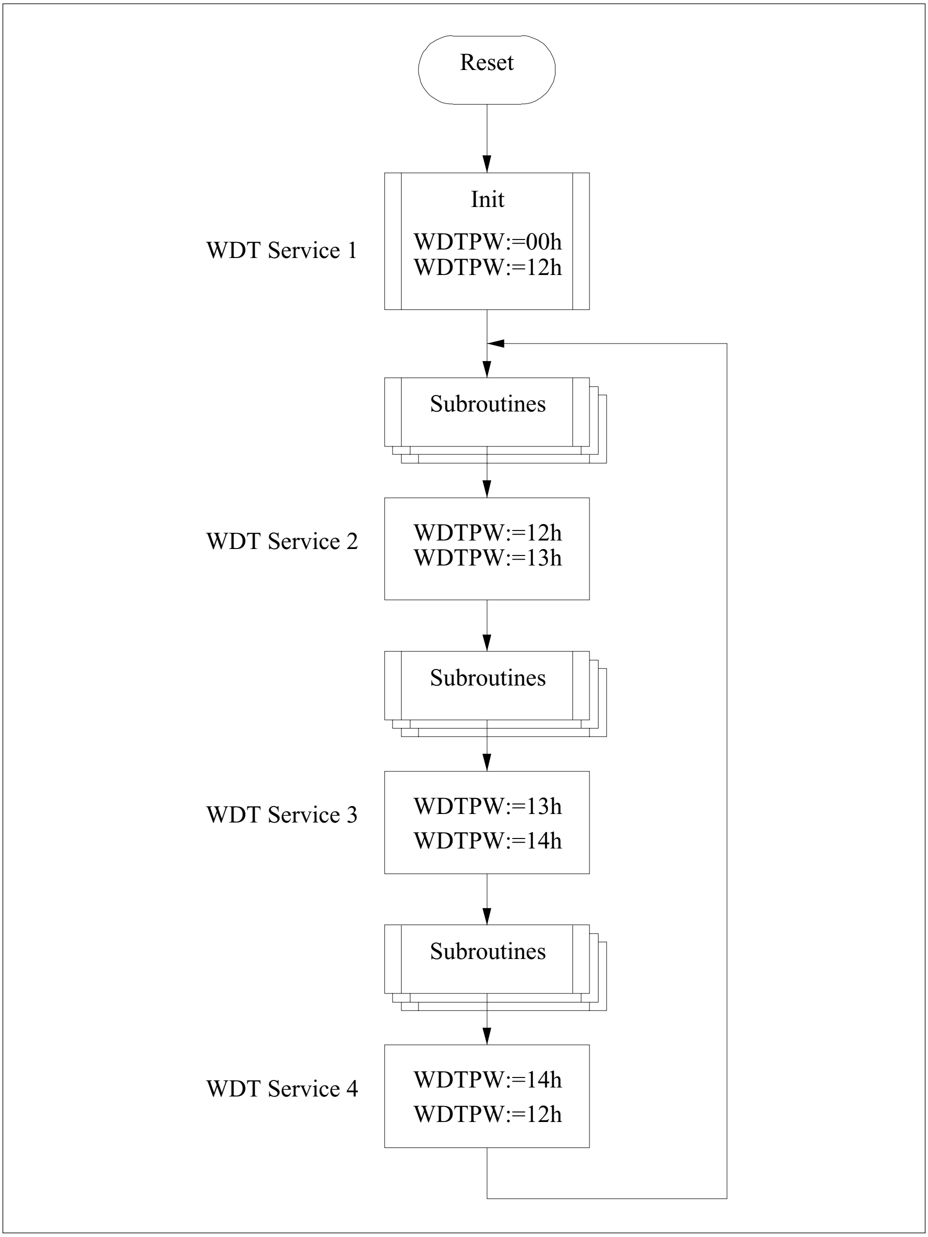


Figure 2 Sequence of Watchdog Timer Services

6.2 Single Service Sequence

Another way is to place only one Watchdog service sequence inside the program, normally at the end of the main loop (see Figure 3). Here, again monitoring of the program flow can be achieved if one uses a temporary variable, TMP, which is modified appropriately at different points during the code execution. At the end of the loop, the content of this variable is used as the WDT password. If everything was OK, the TMP value exactly matches the password value stored in WDTCON0. However, if one or several checkpoints in the code have been missed due to failures, TMP does not contain the correct value and an early Watchdog Time-out will be generated, setting the bit WDTSR.WDTAE. If you want to use this feature in your application you have to use the procedure `WDT_Pass_Secure(Oldpassword)` further below.

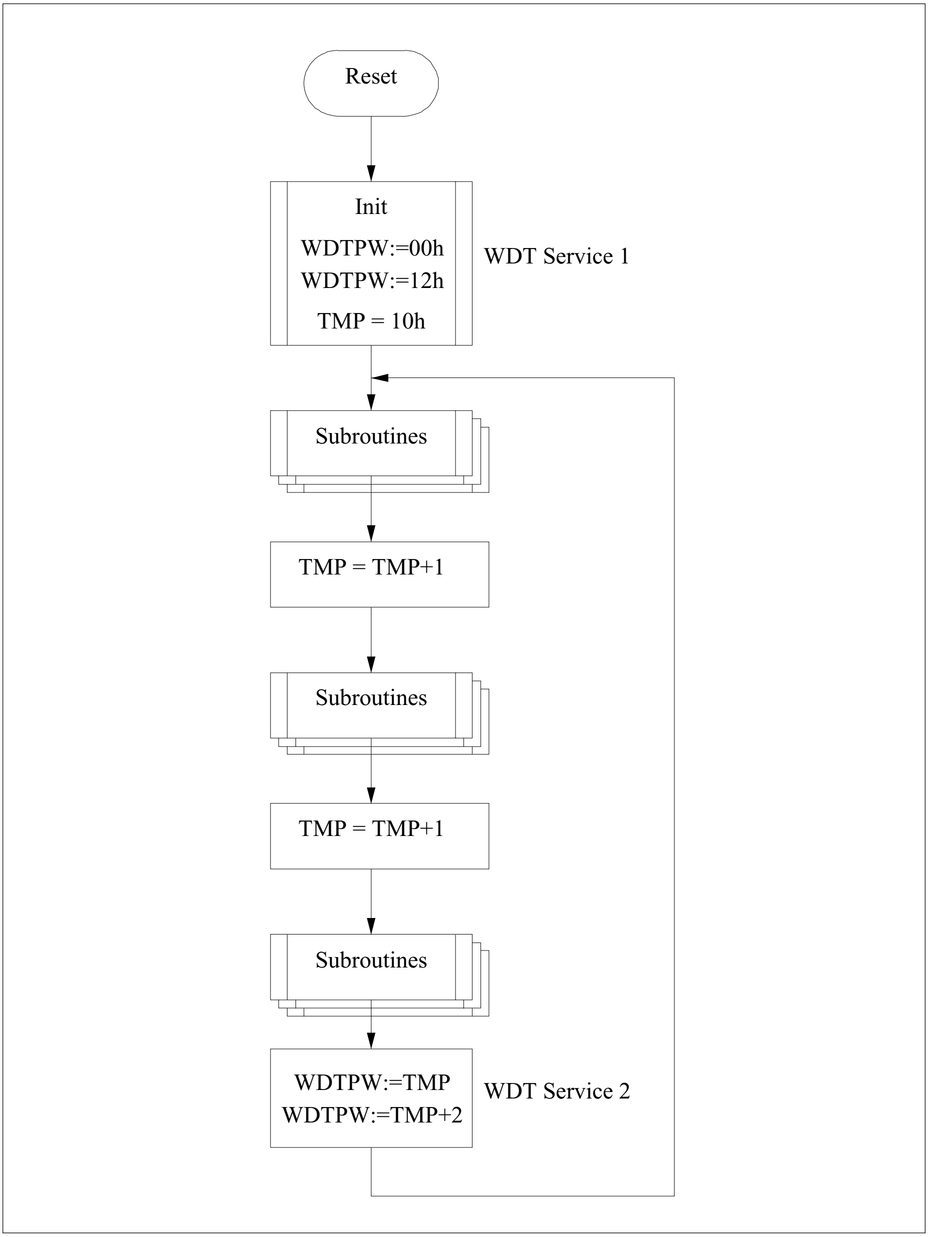


Figure 3 Servicing the WDT at the end of the program loop

6.3 Password Access using the Extended Security Thoughts

The following procedure is nearly the same as the procedure used in chapter Password Access. The great difference is the variable hand over to the procedure. If these variable does not match with the password already stored in WDTCON0.WDTPW a Watchdog Timer Access Error (WDTSR.WDTAE) will occur.

In all error cases, the Watchdog Timer:

- Generates an NMI trap request
- Enters the Reset Pre-warning Mode
- Sets the Watchdog Reset Pre-warning Flag, WDTSR.WDTPR and Watchdog Time-out Period Indication Flag, WDTSR.WDTTO

```
//Password Access to unlock WDTCON0 using the Extended Security Thoughts
//include this in your project
void WDT_Pass_Secure(unsigned char oldpass)
{
  #define WDTCON0 *((volatile unsigned int*) 0xF0000020)
  #define WDTCON1 *((volatile unsigned int*) 0xF0000024)
  unsigned int passwd;
  passwd = WDTCON0;           //load value from WDTCON0 in variable
  passwd = passwd & 0xffff00ff;      //bits from the WDTPW cleared
  passwd = passwd | (oldpass << 8);  //value from variable included
  passwd = passwd | 0x0f0;           //sets bit 7:4 to 1
  passwd =  WDTCON1 & 0x00c)         //bit 2:3 loaded from WDTCON1
        | (passwd & 0xfffffff3);     //and bit 2:3 cleared from WDTCON0
  passwd = (passwd ^ 0x00000002);    //invert bit 1
  WDTCON0 = passwd;  //write password to register WDTCON0
}
```

The following procedure call will unlock the WDTCON0 register if the value of the variable Oldpassword does match with the password already stored in WDTCON0.WDTPW. If not, an error will occur and the WDT will enter in the Reset Pre-warning Mode.

```
//Procedure call
WDT_Pass_Secure(Oldpassword);
```

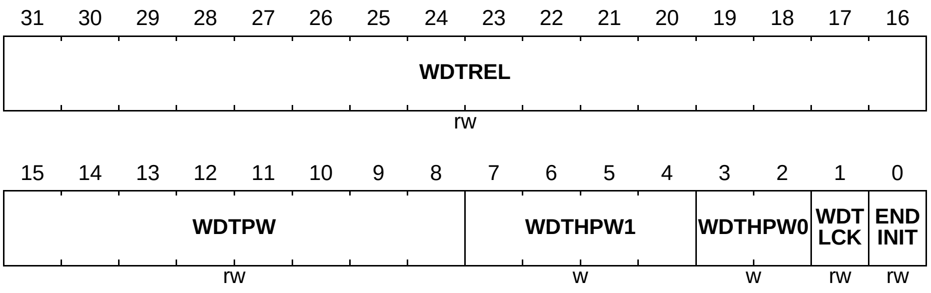
7 WDT Register Overview

Three registers are provided with the Watchdog Timer unit (WDT): WDTCON0, WDTCON1, and WDTSR.

Table 6 WDT Address Map

Address	Name	Description	Block	Read	Write
0xF000 0028	WDTSR	Watchdog Timer Status Register	PWR	U; SV	U; SV; NC
0xF000 0024	WDTCON1	Watchdog Timer Control Register 1	PWR	U; SV	SV; E
0xF000 0020	WDTCON0	Watchdog Timer Control Register 0	PWR	U; SV	SV; PW

WDTCON0
Watchdog Control Register **Reset Value: FFFC 0002$_H$**



Field	Bits	Type	Description
ENDINIT	0	rw	End-of-Initialization Control Bit. 0 Access to Endinit-protected registers is permitted. 1 Access to Endinit-protected registers is not permitted. ENDINIT controls the access to critical system registers. Must be written with the same value during a password access.Can be modified during a modify access to WDTCON0.

WDTLCK	1	rw	Lock bit to Control Access to WDTCON0. 0 Register WDTCON0 is unlocked. 1 Register WDTCON0 is locked. WDTLCK must be reset to 0 during the password access to WDTCON0 and set to 1 during the modify access to WDTCON0 (the inverted value of WDTLCK always must be written to this bit). WDTLCK is controlled by hardware and is cleared after a successful password access to WDTCON0 and automatically set again after a successful modify access to WDTCON0. The value written to this bit location is only used for the protection mechanism and is not stored.
WDTHPW0	[3:2]	w	Hardware Password 0. This field must be written to the value of bits WDTDR and WDTIR in register WDTCON1 during a password access. WDTHPW0 must be written to 00b during a modify access to WDTCON0. When read, these bits always return 0.
WDTHPW1	[7:4]	w	Hardware Password 1. This field must be written to 1111b during both, a password access and a modify access to WDTCON0. When read these bits always return 0.
WDTPW	[15:8]	rw	User-Definable Password Field for Access to WDTCON0. Must be written with the same value during a password access. It can be modified during a modify access to WDTCON0.
WDTREL	[31:16]	rw	Reload Value for the Watchdog Timer. If the Watchdog Timer is enabled, it will start counting from this value after a correct Watchdog service. This field must be written with the same value during a password access. It can be modified during a modify access to WDTCON0.

WDTCON1
Watchdog Control Register 1 **Reset Value: 0000 0002$_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0					**SCKCLR**			**0**				**WDT DR**	**WDT IR**	**0**	
					rw							rw	rw		

Field	Bits	Type	Description
WDTIR	2	rw	Watchdog Timer Input Frequency Request Control Bit. 0 Request to set input frequency to f_{SYSCLK} /16384. 1 Request to set input frequency to f_{SYSCLK} /256. This bit can only be modified if ENDINIT is reset to 0. An update of this bit only goes into effect when ENDINIT is set to 1 again. As long as ENDINIT is 0, bit WDTIS in register WDTSR controls the current input frequency of the Watchdog Timer. When ENDINIT is 1, WDTIS is updated with the state of bit WDTIR.
WDTDR	3	rw	Watchdog Timer Disable Request Control Bit. 0 Request to enable the Watchdog Timer. 1 Request to disable the Watchdog Timer. This bit can only be modified if ENDINIT is reset to 0. An update of this bit only goes into effect when ENDINIT is set to 1 again. As long as ENDINIT is 0, bit WDTDS in register WDTSR controls the current enable/disable status of the Watchdog Timer. When ENDINIT is 1, WDTDS is updated with the state of bit WDTDR.

SCKCLR	[10:8]	rw	Self check clear request bits. SCKCLR = <number>: request to clear bit SCKERR[<number>]. When ENDINIT is set to 1, the corresponding bit SCKERR in register WDTSR will be cleared.
0		r	These bit positions are read-only, returning 0 when read. Writing to these bit positions has no effect. These positions are reserved for future extensions, and it is advised to always write a 0 to these bit positions when writing to the register in order to preserve compatibility with future derivatives.

WDTSR
Watchdog Status Register **Reset Value: FFFC uu1u$_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
WDTTIM															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCKERR								**0**		**WDT PR**	**WDT PO**	**WDT DS**	**WDT IS**	**WDT OE**	**WDT AE**
r										r	r	r	r	r	r

Field	**Bits**	**Type**	**Description**
WDTAE	0	r	Watchdog Access Error Status Flag. 0 No Watchdog access error. 1 A Watchdog access error has occurred. This bit is set by hardware when an illegal access to register WDTCON0 was attempted (either a password or modifying access). This bit is only reset through: - a power-on, hardware, or software reset; - after ENDINIT is written to 1 during a modify access to register WDTCON0 (not possible if Watchdog is in the reset prewarning phase, WDTPR = 1).

WDTOE	1	r	Watchdog Overflow Error Status Flag. 0 No Watchdog overflow error. 1 An Watchdog overflow error has occurred. This bit is set by hardware when the Watchdog Timer is enabled and is not serviced before the overflow from 0xFFFF to 0x0000 occurs, or if the Watchdog Timer is in time-out mode and the overflow occurs. This bit is only reset through: - a power-on, hardware, or software reset; - after ENDINIT is written to 1 during a modify access to register WDTCON0 (not possible if Watchdog is in the reset prewarning Phase, WDTPR = 1).
WDTIS	2	r	Watchdog Input Clock Status Bit. 0 Watchdog Timer input clock is f_{SYSCLK} /16384 (default after reset). 1 Watchdog Timer input clock is f_{SYSCLK} /256. This bit is updated with the state of bit WDTIR after ENDINIT is written to 1 during a modify access to register WDTCON0.
WDTDS	3	r	Watchdog Enable/Disable Bit. 0 Watchdog Timer is enabled (default after reset). 1 Watchdog Timer is disabled. This bit is updated with the state of bit WDTDR after ENDINIT is written to 1 during a modify access to register WDTCON0.

WDTTO	4	r	Watchdog Time-out Period Indication Flag. 0 Normal mode. 1 The Watchdog is operating in time-out mode (default after reset). Time-out mode is entered automatically after a reset and after the first password access to register WDTCON0 or when a Watchdog error is detected. Time-out mode is terminated only in a non-error case with a modify access to WDTCON0 writing ENDINIT to 1. After termination of the time-out mode, WDTTO is cleared through hardware. If time-out mode is entered due to a Watchdog error, this mode cannot be terminated until the Watchdog reset occurs.
WDTPR	5	r	Watchdog Reset Prewarning Flag. 0 Normal mode (default after reset). 1 A Watchdog error has occurred. The Watchdog has issued an NMI trap and is in the final time-out phase (WDTTO is also set in this case). A reset of the chip occurs after the time-out has expired. This bit can be examined in the NMI trap routine to determine the cause of the trap. WDTPR is cleared only through a reset.
SCKERR	[15:8]	r	Self check error bits. SCKERR[<number>] = 0: Self check <number> passed successfully. SCKERR[<number>] = 1: Self check <number> not passed This bits are set to 1 after reset when the self check configuration has been selected. Each bit of this array can be cleared 1 by 1 through the SCKCLR bits in register WDTCON1. This operation will go into effect after ENDINIT is written to 1 during a modify access to register WDTCON0. It is not possible to terminate the self check mode until all SCKERR bits are cleared.

WDTTIM	[31:16]	r	Reflects the current contents of the Watchdog Timer.
0	6, 7	r	These bit positions are read-only, returning 0 when read. Writing to these bit positions has no effect. These positions are reserved for future extensions, and it is advised to always write a 0 to these bit positions when writing to the register in order to preserve compatibility with future derivatives.

Infineon goes for Business Excellence

“Business excellence means intelligent approaches and clearly defined processes, which are both constantly under review and ultimately lead to good operating results.
Better operating results and business excellence mean less idleness and wastefulness for all of us, more professional success, more accurate information, a better overview and, thereby, less frustration and more satisfaction.”

Dr. Ulrich Schumacher

http://www.infineon.com

Published by Infineon Technologies AG