Chapter 1

Real Time Interrupt Module

Topic Page

1.1 Revision History

Date	Version	Author	Description
05/31/02	Rev. A	Frank Noha	Initial revision
06/21/02	Rev. A.1	Frank Noha	changed all references to counter block 1 and 2 to 0 and 1 (i.e. UC2 -> UC1, ...) page 5: changed reference to external clock timout counter to external clock supervising circuit page 11: added note for TBHCOMP and TBLCOMP values updated detection period for NTU page 14: added note for standby mode with PLL off page 56: removed “TBINT flag setting when external timebase counter reaches zero” page 57: clarified setting of TBINT flag section 1.9: updated interface signal list removed VBUSP-FIRST, VBUSP-BYTECNT, VBUSP-AMODE, VBUSP-AERROR changed overflow interrupts and capture events to bus type
08/23/02	Rev. A.2	Frank Noha	changed all references from WKEY to RTIWDKEY page 10: changed incoming signal name Up Counter 1 to Up Counter 0. page 11 end of first paragraph: changed Free Running Counter 1 to Free Running Counter 0. added additional Note, to set RTIGCTRL[0]. page 13: changed NOR gate in Blockdiagram to reflect American symbol page 52: corrected typo for writting 1 to register bits 18, 17, 16 to reflect disabling of interrupts page 59: changed bit setting after Reset to 0xA3 page 62/63: changed signal names with RTI_ prefix page 63: changed WD_RESET signal to RTI_RESETn
08/30/02	Rev. A.3	Frank Noha	page 14: added overflow interrupt generation in standby mode
09/27/02	Rev. A.4	Frank Noha	page 38, 40, 42, 44: changed wording of interrupt generation to flagged and generate DMA request to initiate DMA request changed note to state that the compare will only be active at counter enable
01/22/03	Rev. A.5	Frank Noha	added variable “platform_name” page 14: added chapter 1.7.5 Suspend Mode behaviour

Date	Version	Author	Description
03/21/03	Rev. A.6	Frank Noha	clarified RTITBHCOMP behaviour on page 12 and page 49
06/11/03	Rev. A.7	Frank Noha	updated read of reserved register bits to read as 0 page 11: added paragraph about NTU detection page 14: updated key values to 16 bit page 21: increased size of WDKEY register to 16 bit page 61: updated WDKEY register for new values page 64: added NTU period requirement
06/12/03	Rev. A.8	Frank Noha	page 10: clarified capture event read page 11: modified paragraph about NTU detection page 12: clarified behaviour if NTU pulse is not detected after switching to ext. timebase page 34,39: clarified capture event read page 64: modified NTU period requirements
07/31/03	Rev. A.9	Frank Noha	added Digital Watchdog changed reset value on RTIWDKEY register to be consistent with the DWD spec
08/04/03	Rev. A.10	Frank Noha	page 15: updated blockdiagram for suspend mode behaviour and changed RTICLK to OSC-CLK for DWD Counter page 18: added DWD behaviour in suspend mode page 23: added KEYST bit in RTIWDSTATUS page 65: added KEYST bit and comment about status bits page 70: added OSC_CLK page71: added System Interface section
12/17/03	Rev. A.11	Frank Noha	page 15ff, page 64: changed DWDPRLD value to 12 bits added Digital Watchdog down counter
06/23/04	Rev. A.12	Frank Noha	page 13: corrected typo’s page 56: corrected enabled to disabled in 1st sentence
03/28/05	Rev. A.13	Frank Noha	page 67: changed reset value of WDKEY register

1.2 Reference documents

The version number of a document is related to the version used for the Real Time Interrupt Module development

Reference number	Document Name/Company	Reference	version

1.3 General Description

This document describes the functionality of the Real Time Interrupt Module used on TMS470 Platform derivatives.

1.4 Introduction

The Real Time Interrupt Module (RTI) provides timer functionality for operating systems and for benchmarking code. The Module incorporates several counters, which define the timebases needed for scheduling in the operating system.

This module is specifically designed to fulfill the requirements for OSEK (“**O**ffene Systeme und deren **S**chnittstellen für die **E**lektronik im **K**raftfahrzeug”; “Open Systems and the Corresponding Interfaces for Automotive Electronics”) as well as OSEK/Time compliant operating systems.

The timers also give the ability to benchmark certain areas of code by reading the counter contents at the beginning and the end of the desired code range and calculating the difference between the values.

1.4.1 Main Features

- Two independent counter blocks for generating different timebases
  - Each block consists of
    - One 32 bit prescale counter
    - One 32 bit free running counter
    - Two capture registers for capturing the prescale and free running counter on a special event
- Free running counter 0 can be incremented by either the internal prescale counter or by an external event
- External Clock supervising circuit to switch to internal prescale counter 0, if external clock source fails to increment in a predefined window
- Four configurable compare registers for interrupt generation, working either with counter 0 or counter 1. Compare 0 to 3 can also generate DMA request signals
- Automatic update of all compare registers on compare match to generate periodic interrupts
- Fast enabling/disabling of interrupts
- Analog Watchdog via external RC Network to prevent for runaway code
- Digital Watchdog
- RTI clock input derived from either VBUS or Oscillator output clock, selectable in the System Module

1.5 Register Mapping

Table 1–1. Register Mapping

Offset Address†	Mnemonic	Name	Description	Page
00h	RTIGCTRL	Global Control Register	starts / stops the counters	1-24
04h	RTITBCTRL	Timebase Control	selection which source triggers free running counter 0	1-26
08h	RTICAPCTRL	Capture Control	controls the capture source for the counters	1-28
0Ch	RTICOMPCTRL	Compare Control	controls the source for the compare registers	1-30
10h	RTIFRC0	Free Running Counter 0	current value of free running counter 0	1-32
14h	RTIUC0	Up Counter 0	current value of prescale counter 0	1-33
18h	RTICPUC0	Compare Up Counter 0	compare value compared with prescale counter 0	1-34
20h	RTICAFRC0	Capture Free Running Counter 0	current value of free running counter 0 on external event	1-35
24h	RTICAUC0	Capture Up Counter 0	current value of prescale counter 0 on external event	1-36
30h	RTIFRC1	Free Running Counter 1	current value of free running counter 1	1-37
34h	RTIUC1	Up Counter 1	current value of prescale counter 1	1-38
38h	RTICPUC1	Compare Up Counter 1	compare value compared with prescale counter 1	1-39
40h	RTICAFRC1	Capture Free Running Counter 1	current value of free running counter 1 on external event	1-40
44h	RTICAUC1	Capture Up Counter 1	current value of prescale counter 1 on external event	1-41
50h	RTICOMP0	Compare 0	compare value to be compared with the counters	1-42

Table 1–1. Register Mapping

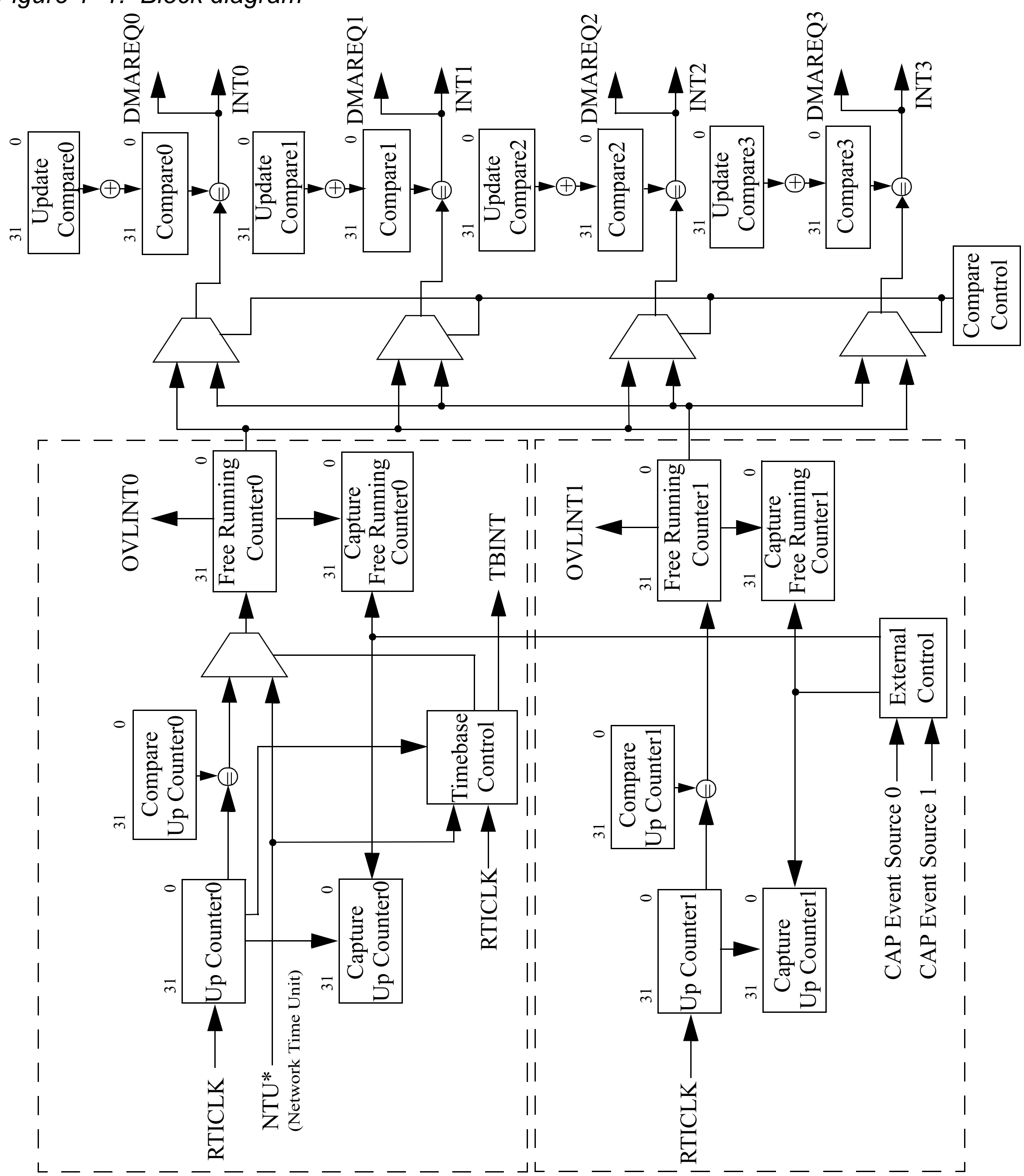
Offset Address†	Mnemonic	Name	Description	Page
54h	RTIUDCP0	Update Compare 0	value to be added to the compare register 0 value on compare match	1-43
58h	RTICOMP1	Compare 1	compare value to be compared with the counters	1-44
5Ch	RTIUDCP1	Update Compare 1	value to be added to the compare register 1 value on compare match	1-45
60h	RTICOMP2	Compare 2	compare value to be compared with the counters	1-46
64h	RTIUDCP2	Update Compare 2	value to be added to the compare register 2 value on compare match	1-47
68h	RTICOMP3	Compare 3	compare value to be compared with the counters	1-48
6Ch	RTIUDCP3	Update Compare 3	value to be added to the compare register 3 value on compare match	1-49
70h	RTITBLCOMP	Timebase Low Compare	compare value to activate edge detection circuit	1-50
74h	RTITBHCOMP	Timebase High Compare	compare value to deactivate edge detection circuit	1-51
80h	RTISETINT	Set Interrupt Enable	sets interrupt enable bits int RTIINTCTRL without having to do a read-modify-write operation	1-52
84h	RTICLEARINT	Clear Interrupt Enable	clears interrupt enable bits int RTIINTCTRL without having to do a read-modify-write operation	1-56
88h	RTIINTFLAG	Interrupt Flags	interrupt pending bits	1-60
90h	RTIDWDCTRL	Digital Watchdog Control	enables the Digital Watchdog	1-63
94h	RTIDWDPRLD	Digital Watchdog Preload	sets the experation time of the Digital Watchdog	1-64
98h	RTIWDSTATUS	Watchdog Status	reflects the status of Analog and Digital Watchdog	1-65

Table 1–1. Register Mapping

Offset Address†	Mnemonic	Name	Description	Page
9Ch	RTIWDKEY	Watchdog Key	correct written key values discharge the external capacitor	1-67
A0h	RTIDWD-CNTR	Digital Watchdog Down Counter	current value of DWD down counter	1-69

1.6 Block diagram

Figure 1–1. Block diagram



1.7 Module Operation

1.7.1 Counter Operation

The two counter blocks provide the same base functionality, whereas counter block 0 has some additional features.

Each block consists of two 32 bit up counters (Up Counter and Free Running Counter). The Up Counter (UCx) is driven by the RTICLK and counts up until the compare value in the Compare Up Counter register (CPUCx) is reached. When the compare matches, the second counter (FRCx), which is a free running counter is incremented. At the same time UCx is reset to zero. If FRCx overflows a interrupt is generated to the vectored interrupt manager.

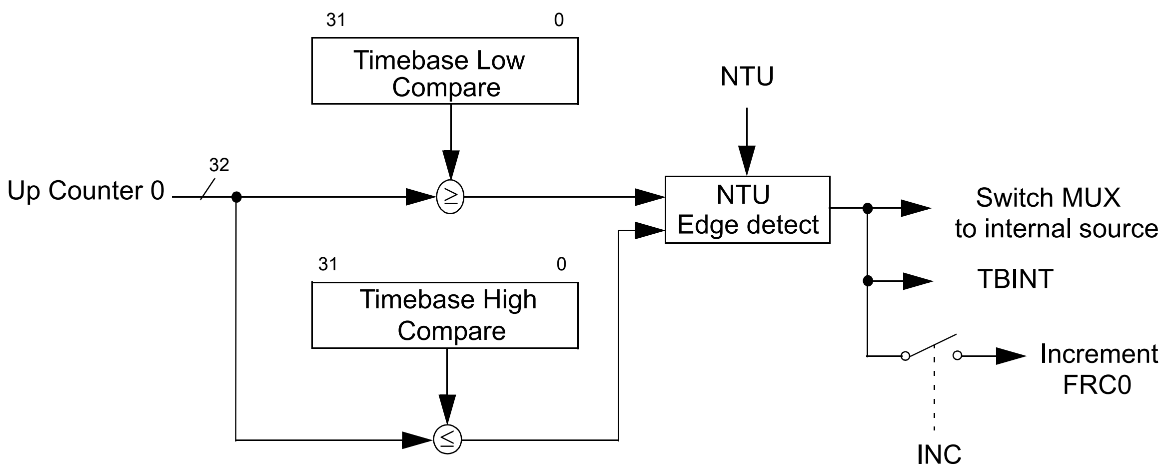
To ensure the consistency of the counters, when both counter value have to be determined, the Free Running Counter has to be read first. This will ensure that at the CPU read cycle, the Up Counter value is stored in the counter register. The second read is done on the Up Counter register, which holds then the value of the counter cycle of the previous read on the Free Running Counter register.

Both blocks provide also a capture feature on external events. Two capture sources can trigger the capture event. Which event triggers block 0 or block 1 is configurable. The sources are coming from the interrupt manager, in order to be able to generate a capture event when one of the peripheral modules has generated an interrupt. The peripheral, which can generate an event is configured in the interrupt manager. When the event is detected, UCx and FRCx are stored in Capture Up Counter (CAUCx) and Capture Free Running Counter (CAFRCx) registers. The read order of the captured values has to be like the order of the actual counters. So the CAFRCx has to be read first and the CAUCx registers has to be read after the CAFRCx value was determined. While the CAFRCx is read the CAUCx value is loaded into a shadow register to ensure data consistency, when during the two reads of the captured data another capture event happens. If the application fails to read the two registers before a second capture event happens, the previous data will be overwritten.

In order to generate interrupt requests to the vectored interrupt manager, there are four compare registers (CPx). Each of the compare registers can be configured to work either on FRC0 or FRC1. When the counter value matches the compare value, an interrupt is generated. All compares provide an additional feature, with an automatic update of the compare value with the value stored in Update Compare (UDCPx) registers when the compare matches. This gives the ability to generate periodic interrupts/DMA requests, without having to update the compare value by software.

1.7.2 Timebase Control

Figure 1–2. Timebase Control



To simplify the operating system design and reduce the need of CPU resources when time triggered functionality is needed, counter block 0 has additional capabilities. The source, which can trigger FRC0 can be either UC0 or an external clock source NTU (i.e. TTCAN, Flexray,...). This can be chosen by software. However to fulfill safety requirements, it is necessary to provide a fallback solution, if the external clock source fails (i.e. due to bus failure). To achieve this, there is a edge detection circuit implemented in the ‘TB Control’ block.

The edge detection of the NTU signal is done with VCLK and the Free Running Counter timebase is synchronized to the RTICLK. To ensure a proper detection of NTU, the NTU period has to be at least twice as long as the RTICLK period and the positive NTU pulsewidth has to be at least two times the VCLK period.

The edge detection circuit will detect clock edges on the NTU signal only when UC0 is bigger or equal than the value stored in Register Timebase Low Compare and lower or equal than the value stored in Register Timebase High Compare. This effectively opens a window in which an edge of the NTU signal is expected. Outside this window, no edges will be detected. If no edge will occure while the edge detection circuit is active, the multiplexer is switched to internal timebase, optionally a timebase interrupt is generated and if the INC bit is set, FRC0 will automatically incremented by one in order to compensate the missed clock cycle of NTU. If an edge occurs, UC0 will be reset in order to synchronize the two timebases. However to synchronize the periods of the two signals, this has to be done by software by adapting CPUC0 to the period of the NTU signal.

If the application decides to switch from internal source to external source a synchronization between the two signals has to be done. This will be done when the TBEXT bit is set. By setting this bit, UC0 will be reset and the edge detection circuit will be active for one (CPUC0 period + TBHCOMP) or until an edge is detected. If there is no pulse during this period the external clock source will be reset to internal clock source. If an edge is detected, the windowed edge detection behaviour will take place. Setting the TBEXT bit will not increment Free Running Counter 0.

If external timebase is used, then software has to ensure that Timebase Low Compare and Timebase High Compare are programmed to a valid state before switching TBEXT to external, in order to alow proper operation of the Timebase Control Circuit. Following condition has to be met: RTITBLCOMP < RTICPUC0 and RTITBHCOMP < RTICPUC0. RTITBHCOMP has to be lower than RTICPUC0, since RTIUC0 will be reset if RTICPUC0 is reached. RTITBHCOMP will represent the number of RTICLK cycles from RTICPUC0 until the circuit switches to internal timebase if no NTU edge is detected.

If external timebase is used, RTIGCTRL[0] has to be set to 1 (enable UC0) in order to ensure that the Timebase Control circuit does not wait indefinitely for an incoming signal.

Figure 1-3 shows a timing example for the the synchronization phase when the TBEXT bit is set.

Figure 1–3. Synchronization example

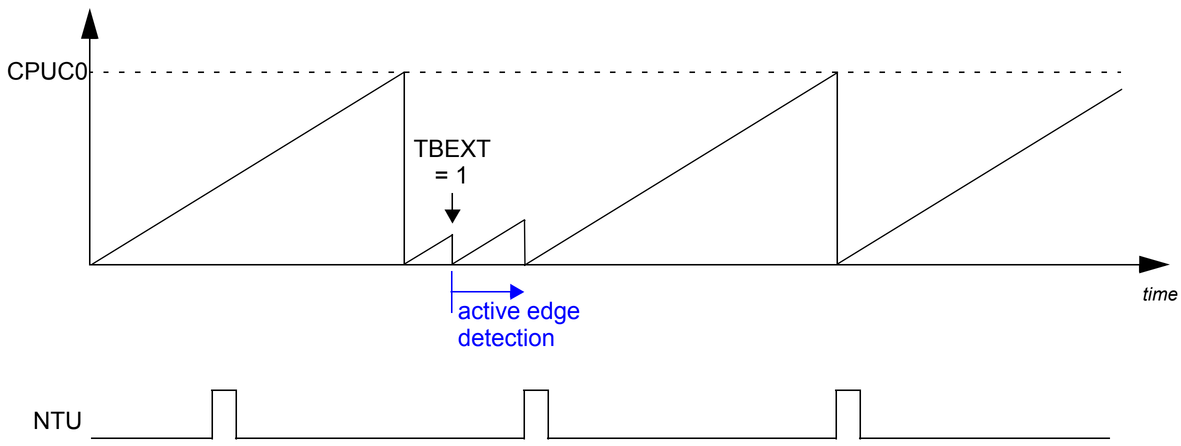
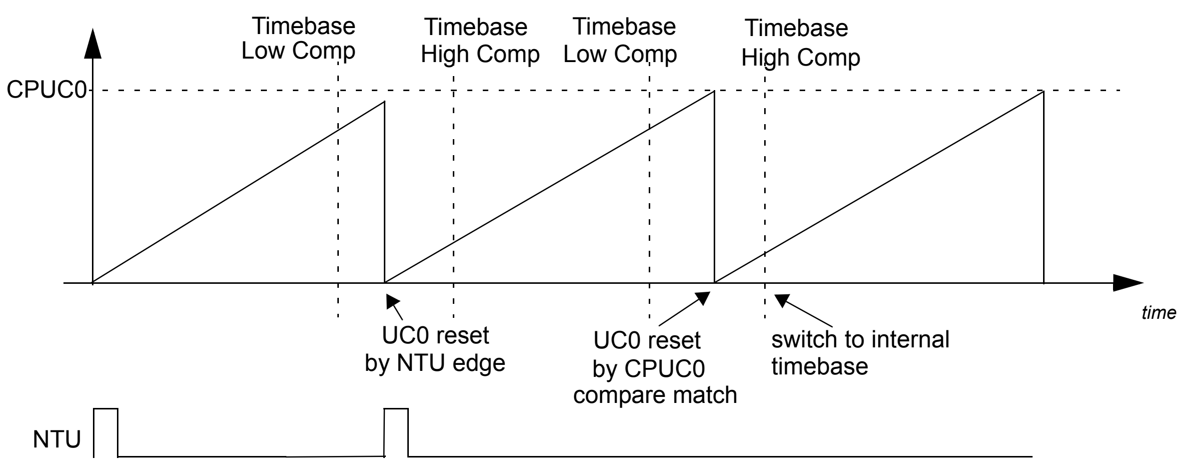
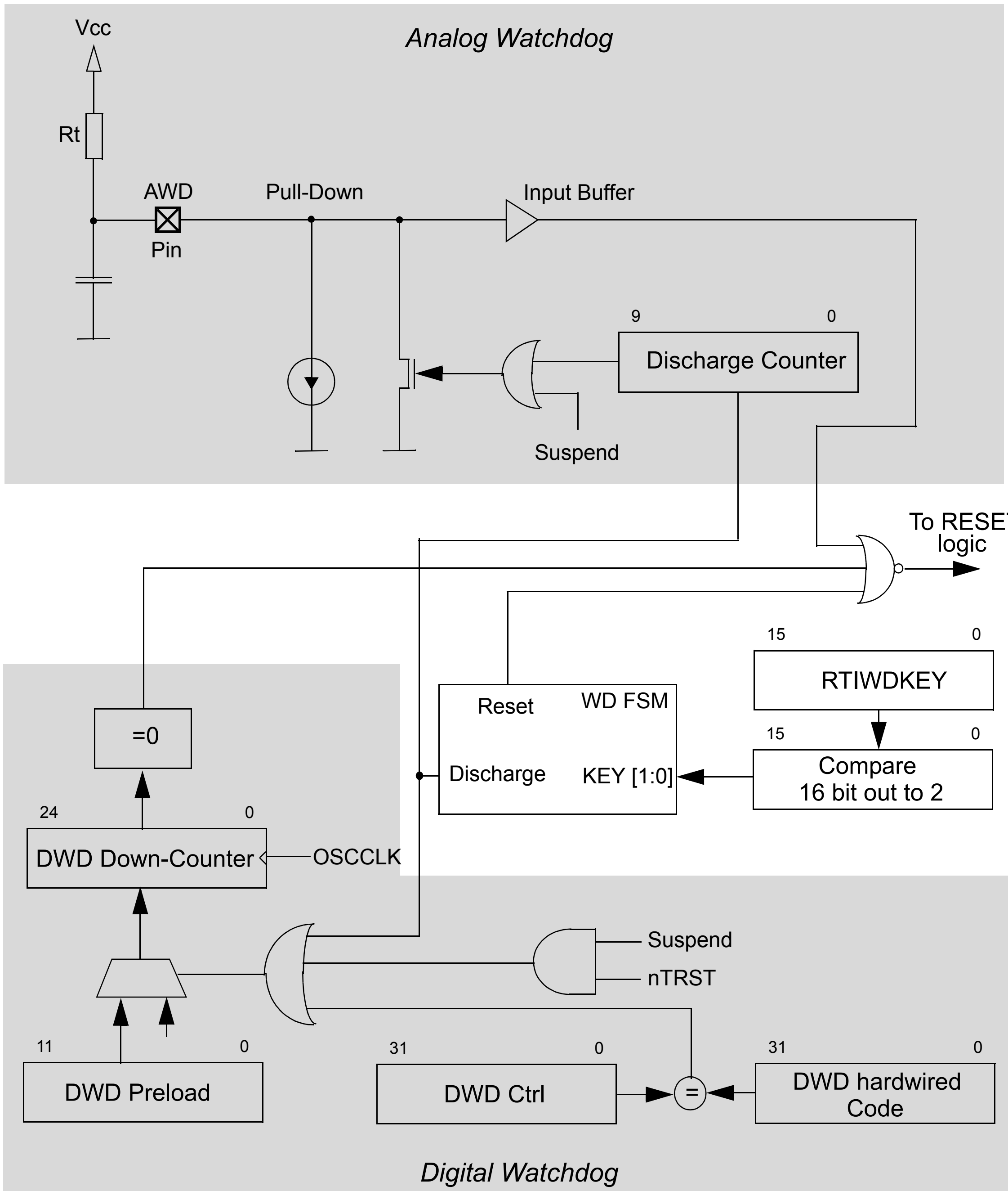


Figure 1–4. Missing NTU signal example



1.7.3 Analog/Digital WatchdogAnalog and Digital Watchdog



The analog and digital watchdogs generate resets to prevent for runaway code. The can be used independently of one or the other.

Analog Watchdog

The periodic of the analog watchdog is determined by an external RC circuit. Whenever the threshold voltage is passed, a reset is generated. The watchdog may be cleared by writing a 0xE51A and then 0xA35C to the RTIWDKEY register. When the correct values are written, the analog watchdog drains the external capacitor and resets the external RC delay. If an incorrect value is written to the RTIWDKEY register, a watchdog reset occurs immediately.

If the pin is left unconnected, the analog watchdog is disabled by the internal pulldown

While the device is in debug mode (Suspend), the external capacitor is always discharged.

The counter which controls the discharge time of the external capacitor should be 10 bit wide. This ensures that the capacitor is discharged correctly.

The values used for the external components have to match the application requirements.

Digital Watchdog

Some applications might want to use a digital watchdog instead of a analog watchdog. The digital watchdog generates also resets after a programmable periode, if no correct key sequence was written to the RTIWDKEY register.

The digital watchdog is disabled per default. If it should be used, it has to be enabled by writing to the RTIDWDCTRL register a 32-bit value which is the inverted value of the hardwired code in the module. Once the DWD is enabled it cannot be disabled.

If the correct key sequence is written to the RTIWDKEY register (0xE51A followed by 0xA35C), the 25-bit DWD Down Counter is reloaded with the 12-bit preload value stored in RTIDWDPRLD. If the incorrect values are written, a watchdog reset will occur immediately. A reset will also be generated, when the DWD Down Counter is decremented to 0.

While the device is in debug mode (Suspend), the DWD Down Counter keeps the value it had when entering debug mode.

The DWD Down Counter will be decremented with OSCCLK frequency.

The experation time of the DWD Down Counter can be determined with following equation:

$$t_{exp} = (RTIDWDPRLD+1) \times 2^{13} / RTICLK \qquad \text{(EQ 1)}$$

where: RTIDWDPRLD = 0...8191

1.7.4 Low Power Mode Operation

The TMS470 Platform can be placed into different Low Power modes (see System Module Specification ‘Low Power Modes’).

The operation of the RTI Module is guaranteed in Run, Idle and Standby mode. In Halt mode all clocks will be switched off and the RTI Module will not work in this mode either.

In Run and Idle modes all parts of the RTI will be active, since it has to be able to wake up the device with Compare and Timebase interrupts. Capturing events generated by the Vectored Interrupt Module is also possible since in both modes the peripheral modules are operational and are able to generate interrupts, which can trigger capture events. It is also possible to generate DMA request in both modes.

In Standby mode the RTI Module will generate Compare, Timebase and Overflow interrupts. The Compare interrupts will periodically wake up the device. The Overflow interrupt will notify the operating system that a counter overflow occured. When the device is put into Standby mode, the peripheral which is generating the external clock NTU is no longer active and the Timebase Control circuitry has to switch to internal clocking scheme, when it detects a missing clock on NTU. The Timebase interrupt will wake up the device and the application software will be able to adapt the periodic interrupt generation to the internal clock source. However DMA requests will not be generated, since the DMA controller will also be powered down. This will prevent DMA transfers after waking up the device, which were probably requested at a time, which does not correlate to the current state of the system. Capturing events while in Standby mode is also not supported.

In the special case of Standby Mode with PLL off, RTICLK might have a different period than with PLL enabled, since RTICLK will be derived from the oscillator output.

Halt Mode will completely deactivate the RTI Module.

1.7.5 Suspend Mode Behaviour

Once the system enters suspend mode, the behaviour of the RTI depends on the COS bit (RTIGCTRL.15). If the bit is cleared and suspend is active, all counters will stop operation. If the bit is set to one, all counters will be clocked normally and the RTI will work like in normal mode. However if the external timebase (NTU) is used and the system is suspended, the timebase control circuit will switch to internal timebase, once it detects the missing NTU signal of the suspended communication controller. This has to be signaled with an TBINT, so that software can resynchronize after suspend is released.

The Analog Watchdog will not be serviced by software in suspend mode. In order to avoid reset generation, the external capacitor will be discharged while in suspend mode.

The DWD counter will not decrement in suspend mode and will hold its current value while in suspend mode.

1.8 Control Registers

1.8.1 Control registers

This section describes the Real Time Interrupt Module registers The registers support 8-bit, 16-bit and 32-bit writes. The offset is relative to the associated peripheral select.

Table 1–2. RTI Registers

Offset Address†	Register	31 15	30 14	29 13	28 12	27 11	26 10	25 9	24 8	23 7	22 6	21 5	20 4	19 3	18 2	17 1	16 0
0x00	RTIGC-TRL	Reserved															
		COS	Reserved													CNT1 EN	CNT0 EN
0x04	RTITBC-TRL	Reserved															
		Reserved														INC	TB EXT
0x08	RTICAPC-TRL	Reserved															
		Reserved														CAP CNTR1	CAP CNTR0
0x0C	RTICOM-PCTRL	Reserved															
		Reserved			COMP SEL3	Reserved			COMP SEL2	Reserved			COMP SEL1	Reserved			COMP SEL0
0x10	RTIFRC0	FRC0 31:0															
0x14	RTIUC0	UC0 31:0															

Table 1–2. RTI Registers (Continued)

Offset Address†	Register	31 / 15	30 / 14	29 / 13	28 / 12	27 / 11	26 / 10	25 / 9	24 / 8	23 / 7	22 / 6	21 / 5	20 / 4	19 / 3	18 / 2	17 / 1	16 / 0
0x18	RTICPUC0	CPUC0 31:0															
0x20	RTICAFRC0	CAFRC0 31:0															
0x24	RTICAUC0	CAUC0 31:0															
0x30	RTIFRC1	FRC1 31:0															
0x34	RTIUC1	UC1 31:0															
0x38	RTICPUC1	CPUC1 31:0															
0x40	RTICAFRC1	CAFRC1 31:0															

Table 1–2. RTI Registers (Continued)

Offset Address†	Register	31 15	30 14	29 13	28 12	27 11	26 10	25 9	24 8	23 7	22 6	21 5	20 4	19 3	18 2	17 1	16 0
0x44	RTICAUC1	CAUC1 31:0															
0x50	RTICOMP0	COMP0 31:0															
0x54	RTIUDCP0	UDCP0 31:0															
0x58	RTICOMP1	COMP1 31:0															
0x5C	RTIUDCP1	UDCP1 31:0															
0x60	RTICOMP2	COMP2 31:0															
0x64	RTIUDCP2	UDCP2 31:0															

Table 1–2. RTI Registers (Continued)

Offset Address†	Register	31 / 15	30 / 14	29 / 13	28 / 12	27 / 11	26 / 10	25 / 9	24 / 8	23 / 7	22 / 6	21 / 5	20 / 4	19 / 3	18 / 2	17 / 1	16 / 0
0x68	RTICOMP3	COMP3 31:0															
0x6C	RTIUDCP3	UDCP3 31:0															
0x70	RTIT-BLCOMP	TBLCOMP 31:0															
0x74	RTITBH-COMP	TBHCOMP 31:0															
0x80	RTISETINT	Reserved													SET OVL1 INT	SET OVL0 INT	SET TBINT
		Reserved				SET DMA3	SET DMA2	SET DMA1	SET DMA0	Reserved				SET INT3	SET INT2	SET INT1	SET INT0
0x84	RTI-CLEAR-INT	Reserved													CLEAR OVL1 INT	CLEAR OVL0 INT	CLEAR TBINT
		Reserved				CLEAR DMA3	CLEAR DMA2	CLEAR DMA1	CLEAR DMA0	Reserved				CLEAR INT3	CLEAR INT2	CLEAR INT1	CLEAR INT0
0x88	RTIINT-FLAG	Reserved													OVL1 INT	OVL0 INT	TBINT
		Reserved												INT3	INT2	INT1	INT0

Table 1–2. RTI Registers (Continued)

<table>
<tr><th>Offset Address†</th><th>Register</th><th>31
15</th><th>30
14</th><th>29
13</th><th>28
12</th><th>27
11</th><th>26
10</th><th>25
9</th><th>24
8</th><th>23
7</th><th>22
6</th><th>21
5</th><th>20
4</th><th>19
3</th><th>18
2</th><th>17
1</th><th>16
0</th></tr>
<tr><td rowspan="2">0x90</td><td rowspan="2">RTID-WDCTRL</td><td colspan="16">DWDCTRL[31:0]</td></tr>
<tr><td colspan="16"></td></tr>
<tr><td rowspan="2">0x94</td><td rowspan="2">RTIDWD-PRLD</td><td colspan="16">Reserved</td></tr>
<tr><td colspan="4">Reserved</td><td colspan="12">DWDPRLD</td></tr>
<tr><td rowspan="2">0x98</td><td rowspan="2">RTIWD-STATUS</td><td colspan="16">Reserved</td></tr>
<tr><td colspan="13">Reserved</td><td>KEYST</td><td>DWDST</td><td>AWDST</td></tr>
<tr><td rowspan="2">0x9C</td><td rowspan="2">RTIWD-KEY</td><td colspan="16">Reserved</td></tr>
<tr><td colspan="16">WDKEY</td></tr>
<tr><td rowspan="2">0xA0</td><td rowspan="2">RTIDWD-CNTR</td><td colspan="7">Reserved</td><td colspan="9">DWDCNTR</td></tr>
<tr><td colspan="16">DWDCNTR</td></tr>
</table>

1.8.2 RTI Global Control Register (RTIGCTRL)

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0x00	Reserved															
	R-0															

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	COS	Reserved													CNT1 EN	CNT0 EN
	RWP-0	R-0													RWP-0	RWP-0

R = Read, W = Write, P = Privilege Mode, U = Undefined; -*n* = Value after reset

Bits 31:16 **Reserved.**

Reads return 0 and writes have no effect

Bit 15 **COS: Continue On Suspend.**

This bit determines if both counters are stopped when the device goes into debug mode or if they continue counting.

User and privilege mode (read):

0 = counters are stopped while in debug mode

1 = counters are running while in debug mode

Privilege mode (write):

0 = stop counters in debug mode

1 = continue counting in debug mode

Bits 14:2 **Reserved.**

Reads return 0 and writes have no effect

Bit 1 **CNT1EN: Counter 1 Enable.**

The CNT1EN bit starts and stops the operation of counter block 1 (UC1 and FRC1).

User and privilege mode (read):

0 = counters are stopped

1 = counters are running

Privilege mode (write):

0 = stop counters

1 = start counters

Bit 0 **CNT0EN: Counter 0 Enable.**

The CNT0EN bit starts and stops the operation of counter block 0 (UC0 and FRC0).

User and privilege mode (read):

0 = counters are stopped

1 = counters are running

Privilege mode (write):

0 = stop counters

1 = start counters

1.8.3 RTI Timebase Control Register (RTITBCTRL)

Bits	31–16
0x04	Reserved
	R-0

Bits	15–2	1	0
	Reserved	INC	TB EXT
	R-0	RWP-0	RWP-0

R = Read, W = Write, P = Privilege Mode, U = Undefined; -*n* = Value after reset

Bits 31:2 **Reserved.**

Reads return 0 and writes have no effect

Bit 1 **INC: Increment Free Running Counter 0.**

This bit determines wether the Free Running Counter 0 is automatically incremented if a failing clock on the NTU signal is detected.

User and privilege mode (read):

0 = FRC0 will not be incremented

1 = FRC0 will be incremented

Privilege mode (write):

0 = Do not increment FRC0 on failing external clock

1 = Increment FRC0 on failing external clock

Bit 0 **TBEXT: Timebase External.**

The Timebase External bit selects whether the Free Running Counter 0 is clocked by the internal Up Counter 0 or from the external signal NTU. Since setting the TBEXT bit to 1 resets Up Counter 0, Free Running Counter 0 will not be incremented in this occurence. The only source which is able to increment Free Running Counter 0 is NTU.

When the Timebase Supervisor circuit detects a missing clockedge, then the TBEXT bit is reset.

The selection if the external signal should be used, can only be done by software.

User and privilege mode (read):

0 = UC0 clocks FRC0

1 = NTU clocks FRC0

Privilege mode (write):

0 = MUX is switched to internal UC0 clocking scheme

1 = MUX is switched to external NTU clocking scheme

1.8.4 RTI Capture Control Register (RTICAPCTRL)

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0x08	Reserved															
	R-0															

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved														CAP CNTR 1	CAP CNTR 0
	R-0														RWP-0	RWP-0

R = Read, W = Write, P = Privilege Mode, U = Undefined; *-n* = Value after reset

Bits 31:2 **Reserved.**

Reads return 0 and writes have no effect

Bit 1 **CAPCNTR1: Capture Counter 1.**

This bit determines, which external interrupt source triggers a capture event of both UC1 and FRC1.

User and privilege mode (read):

0 = capture event is triggered by Capture Event Source 0

1 = capture event is triggered by Capture Event Source 1

Privilege mode (write):

0 = enable capture event triggered by Capture Event Source 0

1 = enable capture event triggered by Capture Event Source 1

Bit 0 **CAPCNTR0: Capture Counter 0.**

This bit determines, which external interrupt source triggers a capture event of both UC0 and FRC0.

User and privilege mode (read):

0 = capture event is triggered by Capture Event Source 0

1 = capture event is triggered by Capture Event Source 1

Privilege mode (write):

0 = enable capture event triggered by Capture Event Source 0

1 = enable capture event triggered by Capture Event Source 1

1.8.5 RTI Compare Control Register (RTICOMPCTRL)

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0x0C	Reserved															
	R-0															

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved			COMPSEL3	Reserved			COMPSEL2	Reserved			COMPSEL1	Reserved			COMPSEL0
	R-0			RWP-0	R-0			RWP-0	R-0			RWP-0	R-0			RWP-0

R = Read, W = Write, P = Privilege Mode, U = Undefined; -*n* = Value after reset

Bits 31:13 **Reserved.**

Reads return 0 and writes have no effect

Bit 12 **COMPSEL3: Compare Select 3.**

This bit determines the counter with which the compare value hold in compare register 3 is compared.

User and privilege mode (read):

0 = value will be compared with FRC 0

1 = value will be compared with FRC 1

Privilege mode (write):

0 = enable compare with FRC 0

1 = enable compare with FRC 1

Bits 11:9 **Reserved.**

Reads return 0 and writes have no effect

Bit 8 **COMPSEL2: Compare Select 2.**

This bit determines the counter with which the compare value hold in compare register 2 is compared.

User and privilege mode (read):

0 = value will be compared with FRC 0

1 = value will be compared with FRC 1

Privilege mode (write):

0 = enable compare with FRC 0

1 = enable compare with FRC 1

Bits 7:5 **Reserved.**

Reads return 0 and writes have no effect

Bit 4 **COMPSEL1: Compare Select 1.**

This bit determines the counter with which the compare value hold in compare register 1 is compared.

User and privilege mode (read):

0 = value will be compared with FRC 0

1 = value will be compared with FRC 1

Privilege mode (write):

0 = enable compare with FRC 0

1 = enable compare with FRC 1

Bits 3:1 **Reserved.**

Reads return 0 and writes have no effect

Bit 0 **COMPSEL0: Compare Select 0.**

This bit determines the counter with which the compare value hold in compare register 0 is compared.

User and privilege mode (read):

0 = value will be compared with FRC 0

1 = value will be compared with FRC 1

Privilege mode (write):

0 = enable compare with FRC 0

1 = enable compare with FRC 1

1.8.6 RTI Free Running Counter 0 Register (RTIFRC0)

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0x10	FRC0 31:0															
	RWP-0															

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RWP-0															

R = Read, W = Write, P = Privilege Mode, U = Undefined; -*n* = Value after reset

Bit 31:0 **FRC0: Free Running Counter 0.**

This registers holds the current value of the Free Running Counter 0 and will be updated continuously.

User and privilege mode (read):

current value of the counter

Privilege mode (write):

the counter can be preset by writing to this register. The counter increments then from this written value upwards.

> If counters have to be preset, they have to be disabled in the RTIGCTRL register in order to ensure consistency between RTIUC0 and RTIFRC0.

1.8.7 RTI Up Counter 0 Register (RTIUC0)

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0x14	UC0 31:0															
	RWP-0															

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RWP-0															

R = Read, W = Write, P = Privilege Mode, U = Undefined; -*n* = Value after reset

Bit 31:0 **UC0: Up Counter 0.**

This registers holds the current value of the Up Counter 0 and prescales the RTI clock. It will be only updated by a previous read of Free Running Counter 0. This gives effectively a 64 bit read of both counters, without having the problem of a counter being updated between two consecutive reads on Up Counter 0 and Free Running Counter 0.

User and privilege mode (read):

value of the counter when the Free Running Counter 0 was read

Privilege mode (write):

the counter can be preset by writing to this register. The counter increments then from this written value upwards.

If counters have to be preset, they have to be disabled in the RTIGCTRL register in order to ensure consistency between RTIUC0 and RTIFRC0.

If the preset value is bigger than the compare value stored in register RTICPUC0 then it can take a long time until a compare matches, since RTIUC0 has to count up until it overflows.

1.8.8 RTI Compare Up Counter 0 Register (RTICPUC0)

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0x18	CPUC0 31:0															
	RWP-0															

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RWP-0															

R = Read, W = Write, P = Privilege Mode, U = Undefined; -*n* = Value after reset

Bit 31:0 **CPUC0: Compare Up Counter 0.**

This registers holds the compare value, which is compared with the Up Counter 0. When the compare matches, Free Running counter 0 is incremented. The Up Counter is set to zero when the counter value matches the CPUC0 value. The value set in this prescales the RTI clock.

$$f_{FRC0} = \frac{RTICLK}{CPUC0 + 1}$$

User and privilege mode (read):

current compare value

Privilege mode (write when TBEXT = 0):

the compare value is updated

Privilege mode (write when TBEXT = 1):

the compare value is not changed

1.8.9 RTI Capture Free Running Counter 0 Register (RTICAFRC0)

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0x20	CAFRC0 31:0															
	R-0															

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R-0															

R = Read, W = Write, P = Privilege Mode, U = Undefined; -*n* = Value after reset

Bit 31:0 **CAFRC0: Capture Free Running Counter 0.**

This registers captures the current value of the Free Running Counter 0 when a event occurs, controlled by the external capture control block.

User and privilege mode (read):

value of Free Running Counter 0 on a capture event

1.8.10 RTI Capture Up Counter 0 Register (RTICAUC0)

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0x24	CAUC0 31:0															
	R-0															

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R-0															

R = Read, W = Write, P = Privilege Mode, U = Undefined; -*n* = Value after reset

Bit 31:0 **CAUC0: Capture Up Counter 0.**

This registers captures the current value of the Up Counter 0 when a event occurs, controlled by the external capture control block. The read sequence has to be the same as with Up Counter 0 and Free Running Counter 0. So the RTICAFRC0 register has to be read first, before the RTICAUC0 register is read. This sequence ensures that the value of the RTICAUC0 register is the corresponding value to the RTICAFRC0 register, even if another capture event happens in between the two reads.

User and privilege mode (read):

value of Up Counter 0 on a capture event

1.8.11 RTI Free Running Counter 1 Register (RTIFRC1)

Bits	***31***	***30***	***29***	***28***	***27***	***26***	***25***	***24***	***23***	***22***	***21***	***20***	***19***	***18***	***17***	***16***
0x30	FRC1 31:0															
	RWP-0															
Bits	***15***	***14***	***13***	***12***	***11***	***10***	***9***	***8***	***7***	***6***	***5***	***4***	***3***	***2***	***1***	***0***
	RWP-0															

R = Read, W = Write, P = Privilege Mode, U = Undefined; -*n* = Value after reset

Bit 31:0 **FRC1: Free Running Counter 1.**

This registers holds the current value of the Free Running Counter 1 and will be updated continuously.

User and privilege mode (read):

current value of the counter

Privilege mode (write):

the counter can be preset by writing to this register. The counter increments then from this written value upwards.

If counters have to be preset, they have to be disabled in the RTIGCTRL register in order to ensure consistency between UC1 and FRC1.

1.8.12 RTI Up Counter 1 Register (RTIUC1)

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0x34	UC1 31:0															
	RWP-0															
Bits	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	RWP-0															

R = Read, W = Write, P = Privilege Mode, U = Undefined; -*n* = Value after reset

Bit 31:0 **UC1: Up Counter 1.**

This registers holds the current value of the Up Counter 1 and prescales the RTI clock. It will be only updated by a previous read of Free Running Counter 1. This gives effectively a 64 bit read of both counters, without having the problem of a counter being updated between two consecutive reads on Up Counter 1 and Free Running Counter 1.

User and privilege mode (read):

value of the counter when the Free Running Counter 1 was read

Privilege mode (write):

the counter can be preset by writing to this register. The counter increments then from this written value upwards.

If counters have to be preset, they have to be disabled in the RTIGCTRL register in order to ensure consistency between UC1 and FRC1.

If the preset value is bigger than the compare value stored in register RTICPUC1 then it can take a long time until a compare matches, since UC1 has to count up until it overflows.

1.8.13 RTI Compare Up Counter 1 Register (RTICPUC1)

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0x38	CPUC1 31:0															
	RWP-0															

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RWP-0															

R = Read, W = Write, P = Privilege Mode, U = Undefined; -*n* = Value after reset

Bit 31:0 **CPUC1: Compare Up Counter 1.**

This registers holds the compare value, which is compared with the Up Counter 1. When the compare matches, Free Running Counter 1 is incremented. The Up Counter is set to zero when the counter value matches the CPUC1 value. The value set in this prescales the RTI clock.

$$f_{FRC1} = \frac{RTICLK}{CPUC1 + 1}$$

User and privilege mode (read):

current compare value

Privilege mode (write):

the compare value is updated

1.8.14 RTI Capture Free Running Counter 1 Register (RTICAFRC1)

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0x40	CAFRC1 31:0															
	R-0															
Bits	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	R-0															

R = Read, W = Write, P = Privilege Mode, U = Undefined; -*n* = Value after reset

Bit 31:0 **CAFRC1: Capture Free Running Counter 1.**

This registers captures the current value of the Free Running Counter 1 when a event occurs, controlled by the external capture control block.

User and privilege mode (read):

value of Free Running Counter 1 on a capture event

1.8.15 RTI Capture Up Counter 1 Register (RTICAUC1)

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0x44	CAUC1 31:0															
	R-0															

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R-0															

R = Read, W = Write, P = Privilege Mode, U = Undefined; -*n* = Value after reset

Bit 31:0 **CAUC1: Capture Up Counter 1.**

This registers captures the current value of the Up Counter 1 when a event occurs, controlled by the external capture control block. The read sequence has to be the same as with Up Counter 1 and Free Running Counter 1. So the RTICAFRC1 register has to be read first, before the RTICAUC1 register is read. This sequence ensures that the value of the RTICAUC0 register is the corresponding value to the RTICAFRC0 register, even if another capture event happens in between the two reads.

User and privilege mode (read):

value of Up Counter 1 on a capture event

1.8.16 RTI Compare 0 Register (RTICOMP0)

Bits	***31***	***30***	***29***	***28***	***27***	***26***	***25***	***24***	***23***	***22***	***21***	***20***	***19***	***18***	***17***	***16***
0x50	COMP0 31:0															
	RWP-0															
Bits	***15***	***14***	***13***	***12***	***11***	***10***	***9***	***8***	***7***	***6***	***5***	***4***	***3***	***2***	***1***	***0***
	RWP-0															

R = Read, W = Write, P = Privilege Mode, U = Undefined; -*n* = Value after reset

Bit 31:0 **COMP0: Compare 0.**

This registers holds a compare value, which is compared with the counter selected in the compare control logic. If the Free Running Counter matches the compare value, an interrupt is flagged. With this register it is also possible to initiate a DMA request.

User and privilege mode (read):

current compare value

Privilege mode (write):

update of the compare register with a new compare value

> A reset does not generate a compare match, since the compare logic will only be active, when the associated counter block is enabled.

1.8.17 RTI Update Compare 0 Register (RTIUDCP0)

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0x54	UDCP0 31:0															
	RWP-0															

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RWP-0															

R = Read, W = Write, P = Privilege Mode, U = Undefined; -*n* = Value after reset

Bit 31:0 **UDCP0: Update Compare 0 Register.**

This registers holds a value, which is added to the value in the compare 0 register each time a compare matches. This gives the possibility to generate periodic interrupts without software intervention.

User and privilege mode (read):

value to be added to the compare 0 register on the next compare match

Privilege mode (write):

new update value

1.8.18 RTI Compare 1 Register (RTICOMP1)

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0x58	COMP1 31:0															
	RWP-0															

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RWP-0															

R = Read, W = Write, P = Privilege Mode, U = Undefined; -*n* = Value after reset

Bit 31:0 **COMP1: Compare 1.**

This registers holds a compare value, which is compared with the counter selected in the compare control logic. If the Free Running Counter matches the compare value, an interrupt is flagged. With this register it is also possible to initiate a DMA request.

User and privilege mode (read):

current compare value

Privilege mode (write):

update of the compare register with a new compare value

A reset does not generate a compare match, since the compare logic will only be active, when the associated counter block is enabled.

1.8.19 RTI Update Compare 1 Register (RTIUDCP1)

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0x5C	UDCP1 31:0															
	RWP-0															

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RWP-0															

R = Read, W = Write, P = Privilege Mode, U = Undefined; -*n* = Value after reset

Bit 31:0 **UDCP1: Update Compare 1 Register.**

This registers holds a value, which is added to the value in the compare 1 register each time a compare matches. This gives the possibility to generate periodic interrupts without software intervention.

User and privilege mode (read):

value to be added to the compare 1 register on the next compare match

Privilege mode (write):

new update value

1.8.20 RTI Compare 2 Register (RTICOMP2)

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0x60	COMP2 31:0															
	RWP-0															

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RWP-0															

R = Read, W = Write, P = Privilege Mode, U = Undefined; -*n* = Value after reset

Bit 31:0 **COMP2: Compare 2.**

This registers holds a compare value, which is compared with the counter selected in the compare control logic. If the Free Running Counter matches the compare value, an interrupt is flagged. With this register it is also possible to initiate a DMA request.

User and privilege mode (read):

current compare value

Privilege mode (write):

update of the compare register with a new compare value

A reset does not generate a compare match, since the compare logic will only be active, when the associated counter block is enabled.

1.8.21 RTI Update Compare 2 Register (RTIUDCP2)

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0x64	UDCP2 31:0															
	RWP-0															

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RWP-0															

R = Read, W = Write, P = Privilege Mode, U = Undefined; -*n* = Value after reset

Bit 31:0 **UDCP2: Update Compare 2 Register.**

This registers holds a value, which is added to the value in the compare 2 register each time a compare matches. This gives the possibility to generate periodic interrupts without software intervention.

User and privilege mode (read):

value to be added to the compare 2 register on the next compare match

Privilege mode (write):

new update value

1.8.22 RTI Compare 3 Register (RTICOMP3)

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0x68	COMP3 31:0															
	RWP-0															
Bits	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	RWP-0															

R = Read, W = Write, P = Privilege Mode, U = Undefined; -*n* = Value after reset

Bit 31:0 **COMP3: Compare 3.**

This registers holds a compare value, which is compared with the counter selected in the compare control logic. If the Free Running Counter matches the compare value, an interrupt is flagged. With this register it is also possible to initiate a DMA request.

User and privilege mode (read):

current compare value

Privilege mode (write):

update of the compare register with a new compare value

A reset does not generate a compare match, since the compare logic will only be active, when the associated counter block is enabled.

1.8.23 RTI Update Compare 3 Register (RTIUDCP3)

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0x6C	UDCP3 31:0															
	RWP-0															

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RWP-0															

R = Read, W = Write, P = Privilege Mode, U = Undefined; -*n* = Value after reset

Bit 31:0 **UDCP3: Update Compare 3 Register.**

This registers holds a value, which is added to the value in the compare 3 register each time a compare matches. This gives the possibility to generate periodic interrupts without software intervention.

User and privilege mode (read):

value to be added to the compare 3 register on the next compare match

Privilege mode (write):

new update value

1.8.24 RTI External Clock Timebase Low Compare Register (RTITBLCOMP)

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0x70	TBLCOMP 31 : 0															
	RWP-0															

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RWP-0															

R = Read, W = Write, P = Privilege Mode, U = Undefined; -*n* = Value after reset

Bit 31:0 **TBLCOMP: Timebase Low Compare Value.**

This value determines when the edge detection circuit starts monitoring the NTU signal. It will be compared with Up Counter 0.

User and privilege mode (read):

current compare value

Privilege mode (write when TBEXT = 0):

the compare value is updated

Privilege mode (write when TBEXT = 1):

the compare value is not changed

A reset does not generate a compare match.

1.8.25 RTI External Clock Timebase High Compare Register (RTITBHCOMP)

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0x74	TBHCOMP 31:0															
	RWP-0															

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RWP-0															

R = Read, W = Write, P = Privilege Mode, U = Undefined; -*n* = Value after reset

Bit 31:0 **TBHCOMP: Timebase High Compare Value.**

This value determines when the edge detection circuit will stop monitoring the NTU signal. It will be compared with Up Counter 0.

RTITBHCOMP has to be less than RTICPUC0, since RTIUC0 will be reset when RTICPUC0 is reached.

Example:

The NTU edge detection circuit should be active +/- 10 RTICLK cycles around RTICPUC0.

RTICPUC0 = 0x00000050
RTITBLCOMP = 0x000046
RTITBHCOMP = 0x00000009

User and privilege mode (read):

current compare value

Privilege mode (write when TBEXT = 0):

the compare value is updated

Privilege mode (write when TBEXT = 1):

the compare value is not changed

A reset does not generate a compare match.

1.8.26 RTI Set/Status Interrupt Register (RTISETINT)

Bits	31–19	18	17	16
0x80	Reserved	SET OVL1 INT	SET OVL0 INT	SET TBINT
	R-0	RWP-0	RWP-0	RWP-0

Bits	15–12	11	10	9	8	7–4	3	2	1	0
	Reserved	SET DMA3	SET DMA2	SET DMA1	SET DMA0	Reserved	SET INT3	SET INT2	SET INT1	SET INT0
	R-0	RWP-0	RWP-0	RWP-0	RWP-0	R-0	RWP-0	RWP-0	RWP-0	RWP-0

R = Read, W = Write, P = Privilege Mode, U = Undefined; *-n* = Value after reset

This register prevents the necessity of a read-modify-write operation if a particular interrupt should be enabled.

Bits 31:19 **Reserved.**

Reads return 0 and writes have no effect

Bit 18 **SETOVL1INT: Set Free Running Counter 1 Overflow Interrupt.**

User and privilege mode (read):

0 = interrupt is disabled

1 = interrupt is enabled

Privilege mode (write):

0 = leaves the corresponding bit unchanged

1 = enable interrupt

Bit 17 **SETOVL0INT: Set Free Running Counter 0 Overflow Interrupt.**

User and privilege mode (read):

0 = interrupt is disabled

1 = interrupt is enabled

Privilege mode (write):

0 = leaves the corresponding bit unchanged

1 = enable interrupt

Bit 16 **SETTBINT: Set Timebase Interrupt.**

User and privilege mode (read):

0 = interrupt is disabled

1 = interrupt is enabled

Privilege mode (write):

0 = leaves the corresponding bit unchanged

1 = enable interrupt

Bits15:12 **Reserved.**

Reads return 0 and writes have no effect

Bit 11 **SETDMA3: Set Compare DMA Request 3.**

User and privilege mode (read):

0 = DMA request is disabled

1 = DMA request is enabled

Privilege mode (write):

0 = leaves the corresponding bit unchanged

1 = enable DMA request

Bit 10 **SETDMA2: Set Compare DMA Request 2.**

User and privilege mode (read):

0 = DMA request is disabled

1 = DMA request is enabled

Privilege mode (write):

0 = leaves the corresponding bit unchanged

1 = enable DMA request

Bit 9 **SETDMA1: Set Compare DMA Request 1.**

User and privilege mode (read):

0 = DMA request is disabled

1 = DMA request is enabled

Privilege mode (write):

0 = leaves the corresponding bit unchanged

1 = enable DMA request

Bit 8 **SETDMA0: Set Compare DMA Request 0.**

User and privilege mode (read):

0 = DMA request is disabled

1 = DMA request is enabled

Privilege mode (write):

0 = leaves the corresponding bit unchanged

1 = enable DMA request

Bits 7:4 **Reserved.**

Reads return 0 and writes have no effect

Bit 3 **SETINT3: Set Compare Interrupt 3.**

User and privilege mode (read):

0 = interrupt is disabled

1 = interrupt is enabled

Privilege mode (write):

0 = leaves the corresponding bit unchanged

1 = enable interrupt

Bit 2 **SETINT2: Set Compare Interrupt 2.**

User and privilege mode (read):

0 = interrupt is disabled

1 = interrupt is enabled

Privilege mode (write):

0 = leaves the corresponding bit unchanged

1 = enable interrupt

Bit 1 **SETINT1: Set Compare Interrupt 1.**

User and privilege mode (read):

0 = interrupt is disabled

1 = interrupt is enabled

Privilege mode (write):

0 = leaves the corresponding bit unchanged

1 = enable interrupt

Bit 0 **SETINT0: Set Compare Interrupt 0.**

User and privilege mode (read):

0 = interrupt is disabled

1 = interrupt is enabled

Privilege mode (write):

0 = leaves the corresponding bit unchanged

1 = enable interrupt

1.8.27 RTI Clear/Status Interrupt Register (RTICLEARINT)

Bits	31–19	18	17	16
0x84	Reserved	CLEAR OVL1 INT	CLEAR OVL0 INT	CLEAR TBINT
	R-0	RWP-0	RWP-0	RWP-0

Bits	15–12	11	10	9	8	7–4	3	2	1	0
	Reserved	CLEAR DMA3	CLEAR DMA2	CLEAR DMA1	CLEAR DMA0	Reserved	CLEAR INT3	CLEAR INT2	CLEAR INT1	CLEAR INT0
	R-0	RWP-0	RWP-0	RWP-0	RWP-0	R-0	RWP-0	RWP-0	RWP-0	RWP-0

R = Read, W = Write, P = Privilege Mode, U = Undefined; -*n* = Value after reset

This register prevents the necessity of a read-modify-write operation if a particular interrupt should be disabled.

Bits 31:19 **Reserved.**

Reads return 0 and writes have no effect

Bit 18 **CLEAROVL1INT: Clear Free Running Counter 1 Overflow Interrupt.**

User and privilege mode (read):

0 = interrupt is disabled

1 = interrupt is enabled

Privilege mode (write):

0 = leaves the corresponding bit unchanged

1 = disable interrupt

Bit 17 **CLEAROVL0INT: Clear Free Running Counter 0 Overflow Interrupt.**

User and privilege mode (read):

0 = interrupt is disabled

1 = interrupt is enabled

Privilege mode (write):

0 = leaves the corresponding bit unchanged

1 = disable interrupt

Bit 16 **CLEARTBINT: Clear Timebase Interrupt.**

User and privilege mode (read):

0 = interrupt is disabled

1 = interrupt is enabled

Privilege mode (write):

0 = leaves the corresponding bit unchanged

1 = disable interrupt

Bits 15:12 **Reserved.**

Reads return 0 and writes have no effect

Bit 11 **CLEARDMA3: Clear Compare DMA Request 3.**

User and privilege mode (read):

0 = DMA request is disabled

1 = DMA request is enabled

Privilege mode (write):

0 = leaves the corresponding bit unchanged

1 = disable DMA request

Bit 10 **CLEARDMA2: Clear Compare DMA Request 2.**

User and privilege mode (read):

0 = DMA request is disabled

1 = DMA request is enabled

Privilege mode (write):

0 = leaves the corresponding bit unchanged

1 = disable DMA request

Bit 9 **CLEARDMA1: Clear Compare DMA Request 1.**

User and privilege mode (read):

0 = DMA request is disabled

1 = DMA request is enabled

Privilege mode (write):

0 = leaves the corresponding bit unchanged

1 = disable DMA request

Bit 8 **CLEARDMA0: Clear Compare DMA Request 0.**

User and privilege mode (read):

0 = DMA request is disabled

1 = DMA request is enabled

Privilege mode (write):

0 = leaves the corresponding bit unchanged

1 = disable DMA request

Bits 7:4 **Reserved.**

Reads return 0 and writes have no effect

Bit 3 **CLEARINT3: Clear Compare Interrupt 3.**

User and privilege mode (read):

0 = interrupt is disabled

1 = interrupt is enabled

Privilege mode (write):

0 = leaves the corresponding bit unchanged

1 = disable interrupt

Bit 2 **CLEARINT2: Clear Compare Interrupt 2.**

User and privilege mode (read):

0 = interrupt is disabled

1 = interrupt is enabled

Privilege mode (write):

0 = leaves the corresponding bit unchanged

1 = disable interrupt

Bit 1 **CLEARINT1: Clear Compare Interrupt 1.**

User and privilege mode (read):

0 = interrupt is disabled

1 = interrupt is enabled

Privilege mode (write):

0 = leaves the corresponding bit unchanged

1 = disable interrupt

Bit 0 **CLEARINT0: Clear Compare Interrupt 0.**

User and privilege mode (read):

0 = interrupt is disabled

1 = interrupt is enabled

Privilege mode (write):

0 = leaves the corresponding bit unchanged

1 = disable interrupt

1.8.28 RTI Interrupt Flag Register (RTIINTFLAG)

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0x88	Reserved													OVL1 INT	OVL0 INT	TBINT
	R-0													RCP-0	RCP-0	RCP-0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved												INT3	INT2	INT1	INT0
	R-0												RCP-0	RCP-0	RCP-0	RCP-0

R = Read, C = Clear, P = Privilege Mode, U = Undefined; -*n* = Value after reset

The corresponding flags are set at every compare match of Free Running Counterx and RTICOMPx value, regardless if the interrupt is enabled or not.

Bits 31:19 **Reserved.**

Reads return 0 and writes have no effect

Bit 18 **OVL1INT: Free Running Counter 1 Overflow Interrupt Flag.**

User and privilege mode (read):

determines if an interrupt is pending

0 = no interrupt pending

1 = interrupt pending

Privilege mode (write):

0 = leaves the bit unchanged

1 = set the bit to 0

Bit 17 **OVL0INT: Free Running Counter 0 Overflow Interrupt Flag.**

User and privilege mode (read):

determines if an interrupt is pending

0 = no interrupt pending

1 = interrupt pending

Privilege mode (write):

0 = leaves the bit unchanged

1 = set the bit to 0

Bit 16 **TBINT: Timebase Interrupt Flag.**

User and privilege mode (read):

this flag is set when the TBEXT bit is cleared by detection of a missing external clockedge. It will not be set by clearing TBEXT by software.

determines if an interrupt is pending

0 = no interrupt pending

1 = interrupt pending

Privilege mode (write):

0 = leaves the bit unchanged

1 = set the bit to 0

Bits 15:4 **Reserved.**

Reads return 0 and writes have no effect

Bit 3 **INT3: Interrupt Flag 3.**

User and privilege mode (read):

determines if a interrupt is pending

0 = no interrupt pending

1 = interrupt pending

Privilege mode (write):

0 = leaves the bit unchanged

1 = set the bit to 0

Bit 2 **INT2: Interrupt Flag 2.**

User and privilege mode (read):

determines if a interrupt is pending

0 = no interrupt pending

1 = interrupt pending

Privilege mode (write):

0 = leaves the bit unchanged

1 = set the bit to 0

Bit 1 **INT1: Interrupt Flag 1.**

User and privilege mode (read):

determines if a interrupt is pending

0 = no interrupt pending

1 = interrupt pending

Privilege mode (write):

0 = leaves the bit unchanged

1 = set the bit to 0

Bit 0 **INT0: Interrupt Flag 0.**

User and privilege mode (read):

determines if a interrupt is pending

0 = no interrupt pending

1 = interrupt pending

Privilege mode (write):

0 = leaves the bit unchanged

1 = set the bit to 0

1.8.29 Digital Watchdog Control Register (RTIDWDCTRL)

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0x90	DWDCTRL[31:0]															
	RWP-0x5312															

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RWP-0xACED															

R = Read, W = Write, P = Privilege Mode, U = Undefined; -*n* = Value after reset

Bits 31:0 **DWDCTRL: Digital Watchdog Control.**

User and priviledge mode (read):

0x5312ACED = DWD counter is disabled

Any other value = DWD counter is enabled

Priviledge mode (write):

By default, the digital watchdog counter is disabled. Any write other than 0x5312ACED to the DWCTRL register enables the counter. This initial write can occur at any time during code execution. Once the initial write has occured, all other writes are ignored. TI recommends that the value 0xACED5312 be written to activate the counter.

Note:

The usage of the RTIDWDCTRL register is to enable/disable once and for all the DWD functionality. The write once register allows this type of functionality. Writing the default value will disable the DWD. Writing an enable value will start the DWD. The write will only be enabled after a reset again.

1.8.30 Digital Watchdog Preload Register (RTIDWDPRLD)

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0x94	Reserved															
	R-0															

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved				DWDPRLD											
	R-0				RWP-0x1FFF											

R = Read, W = Write, P = Privilege Mode, U = Undefined; -*n* = Value after reset

Bits 31:12 **Reserved.**

Bits 11:0 **DWDPRLD: Digital Watchdog Preload Value.**

User and priviledge mode (read):

Current preload value.

Priviledge mode (write):

The preload value must be written prior to enabling the DWD Down Counter. The experation time of the counter can be calculated with the formula given in equation (EQ 1) on page 1-17.

1.8.31 Watchdog Status Register (RTIWDSTATUS)

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0x98	Reserved															
	R-0															

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved													KEYST	DWDST	AWDST
	R-0													RCP-x	RCP-x	RCP-x

R = Read, W = Write, P = Privilege Mode, U = Undefined; -*n* = Value after reset

The values of the following status bits will not be affected by a reset. Only the user can clear these bits. These bits are only intended for debug purposes.

Bits 31:3 **Reserved.**

Bit 2 **KEYST: Watchdog KeyStatus.**

This bit denotes a reset generated by a wrong key or key sequence written to the RTIWDKEY register.

User and priviledge mode (read):

0 = no reset generated

1 = reset generated

Priviledge mode (write):

0 = leaves the current value unchanged

1 = sets the bit to 0

Bit 1 **DWDST: Digital Watchdog Status.**

User and priviledge mode (read):

0 = no reset generated

1 = reset generated

Priviledge mode (write):

0 = leaves the current value unchanged

1 = sets the bit to 0

Bit 0 **AWDST: Analog Watchdog Status.**

User and priviledge mode (read):

0 = no reset generated

1 = reset generated

Priviledge mode (write):

0 = leaves the current value unchanged

1 = sets the bit to 0

1.8.32 Watchdog Key Register (RTIWDKEY)

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0x9C	Reserved															
	R-0															

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	WDKEY															
	RW-0xA35C															

R = Read, W = Write, P = Privilege Mode, U = Undefined; -*n* = Value after reset

Bits 31:8 **Reserved.**

Reads return 0 and writes have no effect

Bit 7:0 **WDKEY: Watchdog Key.**

Key Sequence location.

User and privilege mode (read):

Reads are indeterminate.

Privilege mode (write):

A write of 0xE51A followed by 0xA35C in two separate write operations defines the Key Sequence and discharges the watchdog capacitor. Writing any other

value causes a digital watchdog reset, as shown in Table 1-3.

Table 1–3. Example of a WDKEY sequence

Step	Value written to WDKEY	Result
1	0x0A35C	No Action
2	0x0A35C	No Action
3	0x0E51A	WDKEY is enabled for reset by next 0x0A35C
4	0x0E51A	WDKEY is enabled for reset by next 0x0A35C
5	0x0E51A	WDKEY is enabled for reset by next 0x0A35C
6	0x0A35C	Watchdog is reset
7	0x0A35C	No Action
8	0x0E51A	WDKEY is enabled for reset by next 0x0A35C

Step	Value written to WDKEY	Result
9	0x0A35C	Watchdog is reset
10	0x0E51A	WDKEY is enabled for reset by next 0x0A35C
11	0x02345	System reset; incorrect value written to WDKEY

1.8.33 Digital Watchdog Down Counter (RTIDWDCNTR)

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0xA0	Reserved							DWDCNTR								
	R-0							R-0								

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DWDCNTR															
	R-0															

R = Read, W = Write, P = Privilege Mode, U = Undefined; -*n* = Value after reset

Bits 31:25 **Reserved.**

Reads return 0 and writes have no effect

Bit 24:0 **DWDCNTR: Digital Watchdog Down Counter.**

User and privilege mode (read):

Reads return the current counter value.

Privilege mode (write):

Writes don’t have an effect.

1.9 Interface

1.9.1 Peripheral interface (VBUS)

The RTI is connected to one of the VBUSP port of the PCR.

Table 1–4. VBUS interface

Signal Name	I/O	Bus	Description
vbusp_clk	I	-	vbusp domain clock
vbusp_req	I	-	vbusp register request from the A2V bridge. Active High
vbusp_address	I	9:0	Address from the A2V bridge
vbusp_dir	I	-	Direction of the transaction 0 = write 1 = read
vbusp_bytecnt	I	9:0	Indicates the number of byte for the requested transaction. Decremented as the transfer progresses. Supported values are: 0x1,0x2,0x4
vbusp_wdata	I	31:0	Write data bus
vbusp_prot	I	-	Privilege of the transaction 0 = user mode 1 = privilege mode
vbusp_rdata	O	31:0	Read data bus
vbusp_wready	O	-	Active high signal indicates the completion of the current write data phase.
vbusp_rready	O	-	Active high signal indicates the completion of the current read data phase.
SUSPEND	I	-	Suspend signal.

1.9.2 Interrupt interface

All interrupts generated by the RTI are connected to the VIM.

Table 1–5. VIM interface

Signal Name	I/O	Bus	Description
RTI_INT_req	O	3:0	Interrupt request to VIM Each bit is active high.
RTI_TBINT_req	O	-	Interrupt request to VIM
RTI_OVL_req	O	1:0	Interrupt request to VIM from Free Running Counter 0 and 1. Active high.
RTI_CAPEVT	I	1:0	Capture Event 0 and 1. Active high.

1.9.3 DMA interface

All DMA request generated by the RTI are connected to the DMA.

Table 1–6. DMA interface

Signal Name	I/O	Bus	Description
RTI_DMA_req	O	3:0	DMA request to DMA Each bit is a pulse active high.

1.9.4 Clock Module interface

The RTI receives the clock from the clock module.

Table 1–7. Clock Module interface

Signal Name	I/O	Bus	Description
RTI_CLK	I	-	RTI clock from Clock module
RTI_SYNC	I	-	synchronizes between RTICLK and Oscillator output
OSC_SYNC	I	-	Digital watchdog clock enable

1.9.5 External Clock interface

The RTI receives the external clock from the TT-CAN or Flexray module.

The NTU period has to be at least twice as long as the RTICLK period and the positive pulse of the NTU signal has to be at least two times as long as the VCLK period.

Table 1–8. External Clock interface

Signal Name	I/O	Bus	Description
NTU	I	-	Network Time Unit from peripheral

1.9.6 Watchdog Interface

The RTI connects to the watchdog open drain pad and send the reset signal to the system module.

Table 1–9. Analog watchdog interface

Signal Name	I/O	Bus	Description
RTI_DISn	O	-	Watchdog discharge control
RTI_PIN	I	-	Input from Watchdog pin.
RTI_RESETn	O	-	Watchdog reset to system module. This is an active low signal

Table 1–10. Digital watchdog interface

Signal Name	I/O	Bus	Description
nTRST	I	-	Reset signal from the JTAG TAP controller

1.9.7 System Interface

Table 1–11. System interface

Signal Name	I/O	Bus	Description
SYS_nPORST	I	-	Active Low Power-on system reset.

1.10 Design Considerations

RTICLK and VBUS CLK are two asynchronous clock domains. Special circuitry is needed to ensure correct read operation (operates at VBUS CLK) from registers that are updated by RTICLK.