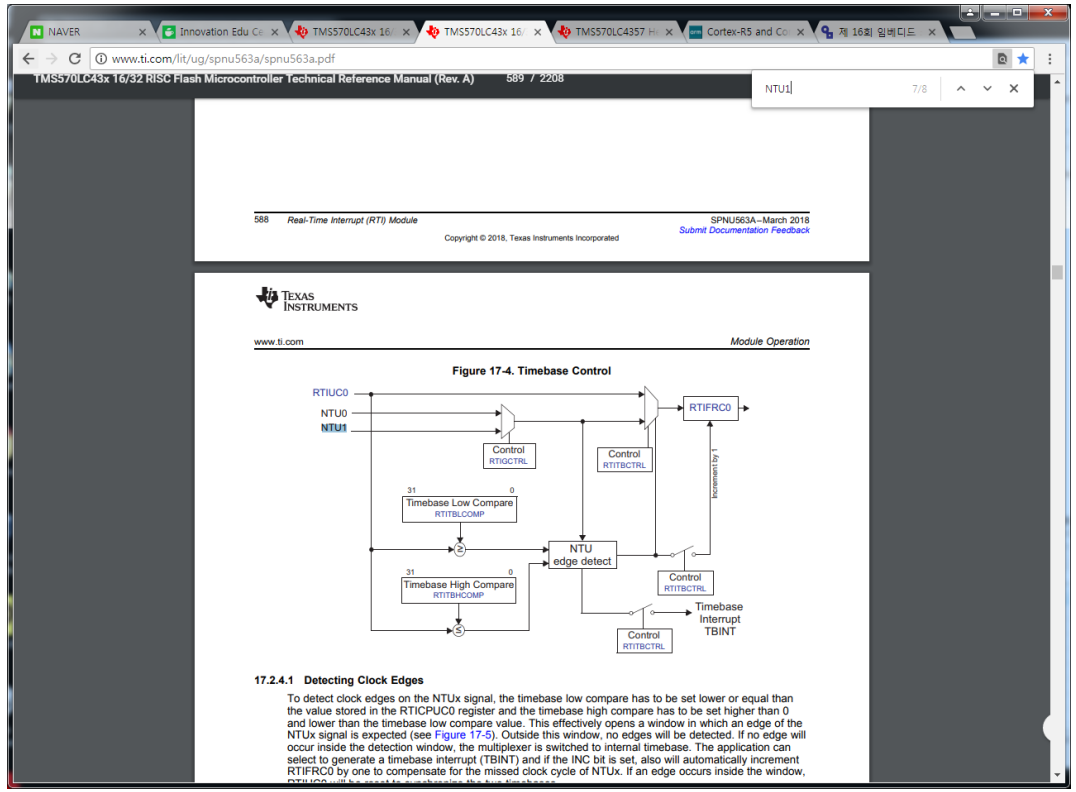
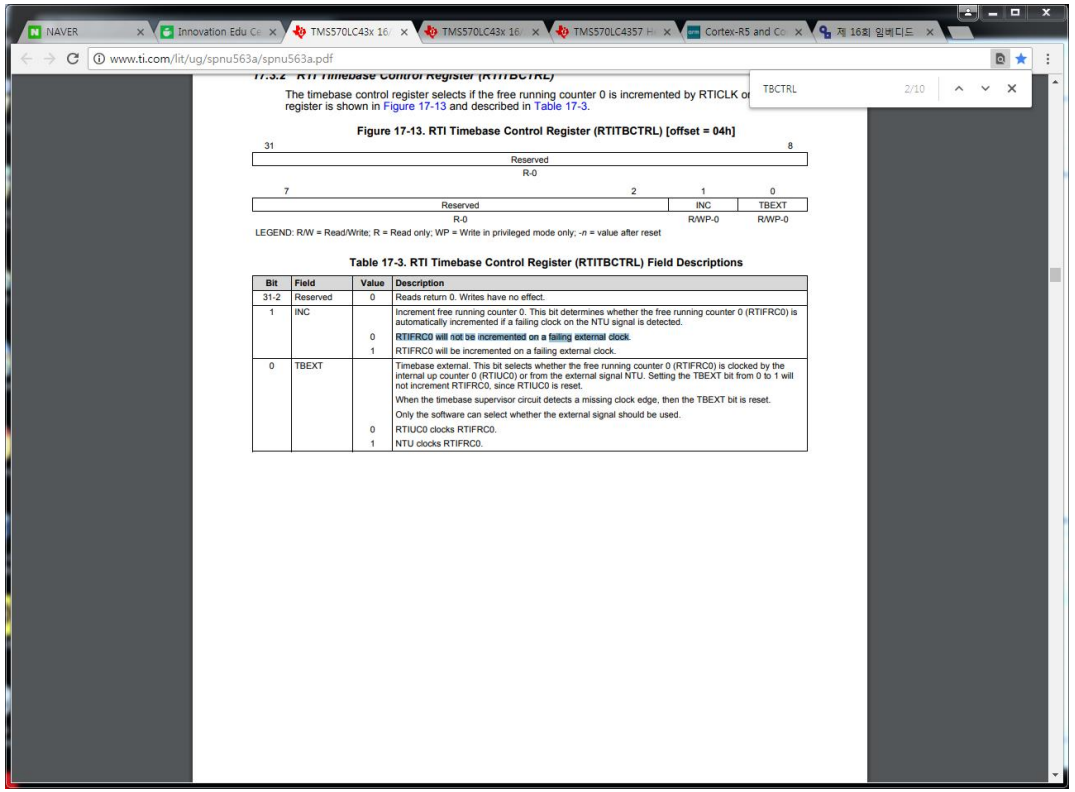
TI DSP, MCU 및 Xilinx Zynq FPGA 프로그램 전문가 과정

Innova Lee(이상훈) 강사
gcccompil3r@gmail.com

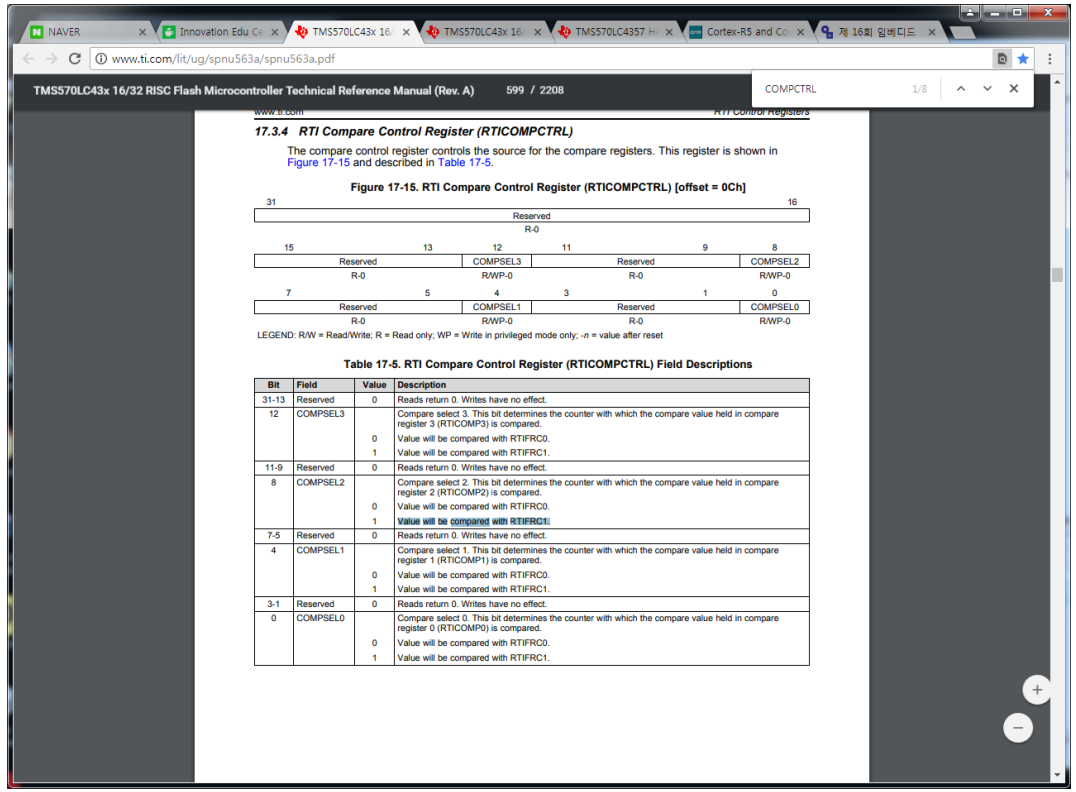
은태영 학생
zero_bird@naver.com

rtilnit()


www.ti.com/lit/ug/spnu563a/spnu563a.pdf
TMS570LC43x 16/32 RISC Flash Microcontroller Technical Reference Manual (Rev. A) 589 / 2208
NTU1 7/8
588 Real-Time Interrupt (RTI) Module SPNU563A–March 2018
Submit Documentation Feedback
Copyright © 2018, Texas Instruments Incorporated
TEXAS INSTRUMENTS
www.ti.com Module Operation
Figure 17-4. Timebase Control
RTIUC0
NTU0
NTU1
Control
RTIGCTRL
Control
RTITBCTRL
RTIFRC0
Increment by 1
31 0
Timebase Low Compare
RTITBLCOMP
NTU
edge detect
31 0
Timebase High Compare
RTITBHCOMP
Control
RTITBCTRL
Timebase
Interrupt
TBINT
Control
RTITBCTRL
17.2.4.1 Detecting Clock Edges
To detect clock edges on the NTUx signal, the timebase low compare has to be set lower or equal than the value stored in the RTICPUC0 register and the timebase high compare has to be set higher than 0 and lower than the timebase low compare value. This effectively opens a window in which an edge of the NTUx signal is expected (see Figure 17-5). Outside this window, no edges will be detected. If no edge will occur inside the detection window, the multiplexer is switched to internal timebase. The application can select to generate a timebase interrupt (TBINT) and if the INC bit is set, also will automatically increment RTIFRC0 by one to compensate for the missed clock cycle of NTUx. If an edge occurs inside the window,


www.ti.com/lit/ug/spnu563a/spnu563a.pdf
TBCTRL 2/10
The timebase control register selects if the free running counter 0 is incremented by RTICLK o
register is shown in Figure 17-13 and described in Table 17-3.
Figure 17-13. RTI Timebase Control Register (RTITBCTRL) [offset = 04h]
31 8
Reserved
R-0
7 2 1 0
Reserved INC TBEXT
R-0 R/WP-0 R/WP-0
LEGEND: R/W = Read/Write; R = Read only; WP = Write in privileged mode only; -n = value after reset
Table 17-3. RTI Timebase Control Register (RTITBCTRL) Field Descriptions
Bit Field Value Description
31-2 Reserved 0 Reads return 0. Writes have no effect.
1 INC Increment free running counter 0. This bit determines whether the free running counter 0 (RTIFRC0) is automatically incremented if a failing clock on the NTU signal is detected.
0 RTIFRC0 will not be incremented on a failing external clock.
1 RTIFRC0 will be incremented on a failing external clock.
0 TBEXT Timebase external. This bit selects whether the free running counter 0 (RTIFRC0) is clocked by the internal up counter 0 (RTIUC0) or from the external signal NTU. Setting the TBEXT bit from 0 to 1 will not increment RTIFRC0, since RTIUC0 is reset.
When the timebase supervisor circuit detects a missing clock edge, then the TBEXT bit is reset.
Only the software can select whether the external signal should be used.
0 RTIUC0 clocks RTIFRC0.
1 NTU clocks RTIFRC0.

rtiInit()



TMS570LC43x 16/32 RISC Flash Microcontroller Technical Reference Manual (Rev. A) 599 / 2208

COMPCTRL 1/8

www.ti.com/lit/ug/spnu563a/spnu563a.pdf

www.ti.com RTI Control Registers

17.3.4 RTI Compare Control Register (RTICOMPCTRL)

The compare control register controls the source for the compare registers. This register is shown in Figure 17-15 and described in Table 17-5.

Figure 17-15. RTI Compare Control Register (RTICOMPCTRL) [offset = 0Ch]

31	16
Reserved	
R-0	

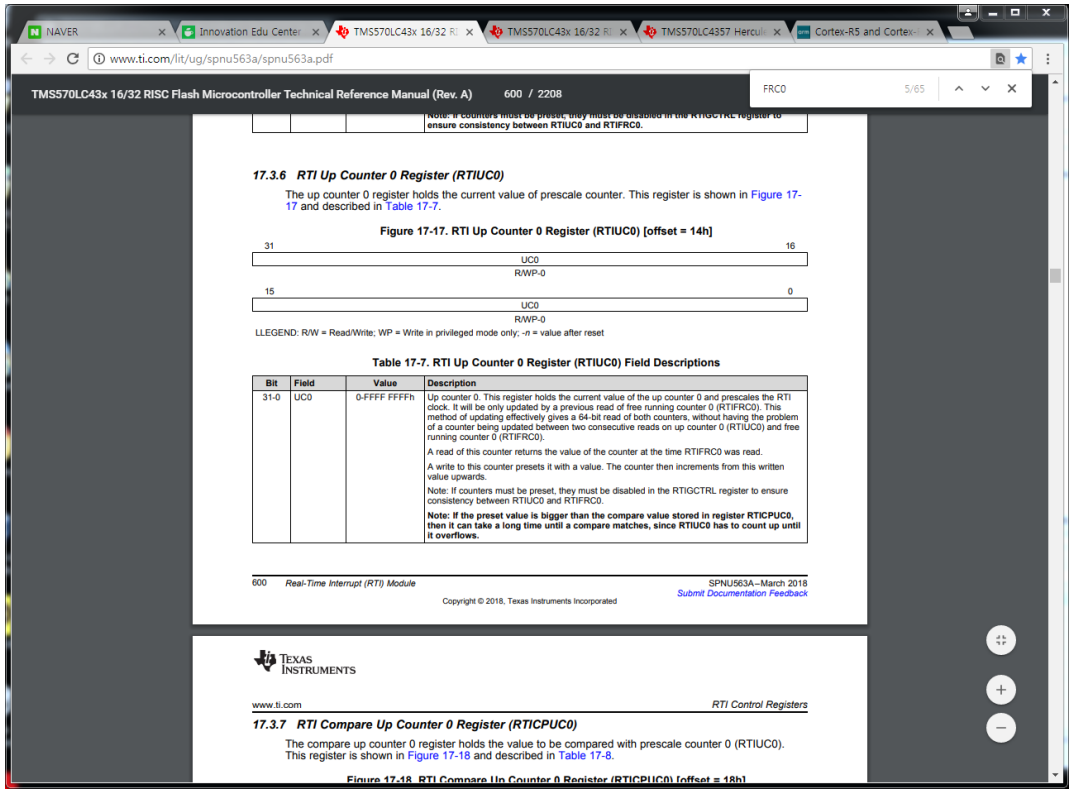
15	13	12	11	9	8
Reserved		COMPSEL3	Reserved		COMPSEL2
R-0		R/WP-0	R-0		R/WP-0

7	5	4	3	1	0
Reserved		COMPSEL1	Reserved		COMPSEL0
R-0		R/WP-0	R-0		R/WP-0

LEGEND: R/W = Read/Write; R = Read only; WP = Write in privileged mode only; -n = value after reset

Table 17-5. RTI Compare Control Register (RTICOMPCTRL) Field Descriptions

Bit	Field	Value	Description
31-13	Reserved	0	Reads return 0. Writes have no effect.
12	COMPSEL3		Compare select 3. This bit determines the counter with which the compare value held in compare register 3 (RTICOMP3) is compared.
		0	Value will be compared with RTIFRC0.
		1	Value will be compared with RTIFRC1.
11-9	Reserved	0	Reads return 0. Writes have no effect.
8	COMPSEL2		Compare select 2. This bit determines the counter with which the compare value held in compare register 2 (RTICOMP2) is compared.
		0	Value will be compared with RTIFRC0.
		1	Value will be compared with RTIFRC1.
7-5	Reserved	0	Reads return 0. Writes have no effect.
4	COMPSEL1		Compare select 1. This bit determines the counter with which the compare value held in compare register 1 (RTICOMP1) is compared.
		0	Value will be compared with RTIFRC0.
		1	Value will be compared with RTIFRC1.
3-1	Reserved	0	Reads return 0. Writes have no effect.
0	COMPSEL0		Compare select 0. This bit determines the counter with which the compare value held in compare register 0 (RTICOMP0) is compared.
		0	Value will be compared with RTIFRC0.
		1	Value will be compared with RTIFRC1.



TMS570LC43x 16/32 RISC Flash Microcontroller Technical Reference Manual (Rev. A) 600 / 2208

FRC0 5/65

www.ti.com/lit/ug/spnu563a/spnu563a.pdf

Note: If counters must be preset, they must be disabled in the RTIGCTRL register to ensure consistency between RTIUC0 and RTIFRC0.

17.3.6 RTI Up Counter 0 Register (RTIUC0)

The up counter 0 register holds the current value of prescale counter. This register is shown in Figure 17-17 and described in Table 17-7.

Figure 17-17. RTI Up Counter 0 Register (RTIUC0) [offset = 14h]

31	16
UC0	
R/WP-0	

15	0
UC0	
R/WP-0	

LLEGEND: R/W = Read/Write; WP = Write in privileged mode only; -n = value after reset

Table 17-7. RTI Up Counter 0 Register (RTIUC0) Field Descriptions

Bit	Field	Value	Description
31-0	UC0	0-FFFF FFFFh	Up counter 0. This register 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 (RTIFRC0). This method of updating effectively gives a 64-bit read of both counters, without having the problem of a counter being updated between two consecutive reads on up counter 0 (RTIUC0) and free running counter 0 (RTIFRC0). A read of this counter returns the value of the counter at the time RTIFRC0 was read. A write to this counter presets it with a value. The counter then increments from this written value upwards. Note: If counters must be preset, they must be disabled in the RTIGCTRL register to ensure consistency between RTIUC0 and RTIFRC0. **Note: 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.**

TEXAS INSTRUMENTS

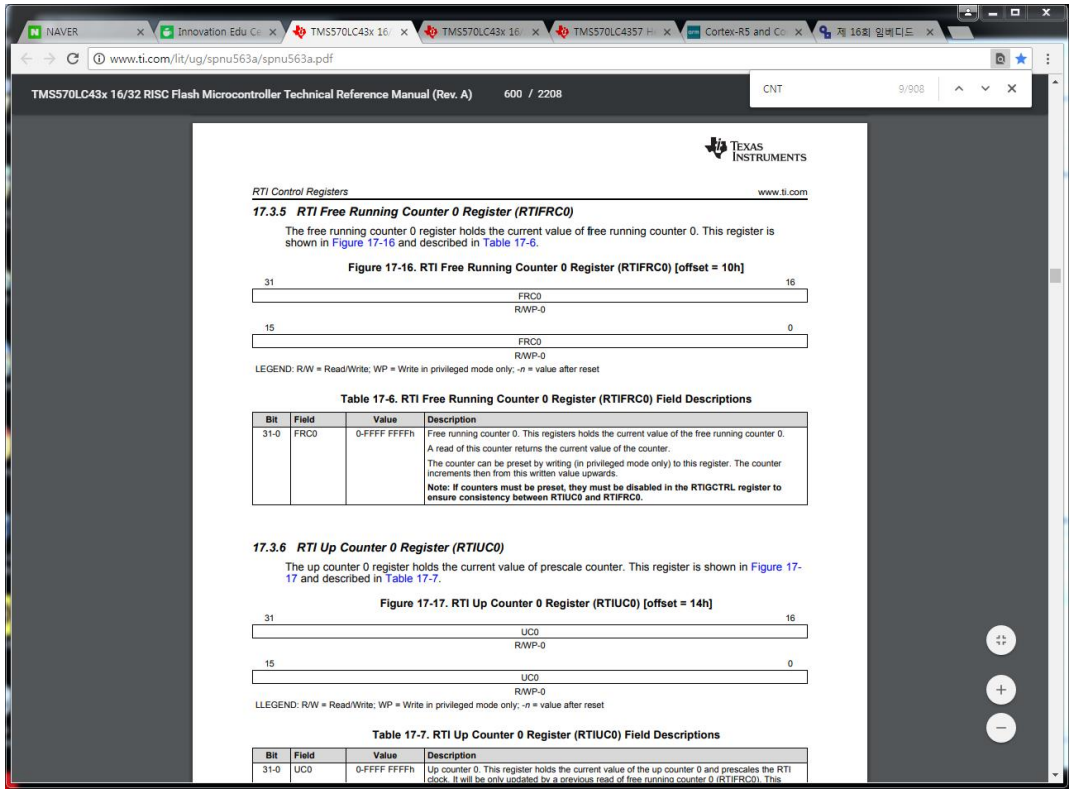
www.ti.com RTI Control Registers

17.3.7 RTI Compare Up Counter 0 Register (RTICPUC0)

The compare up counter 0 register holds the value to be compared with prescale counter 0 (RTIUC0). This register is shown in Figure 17-18 and described in Table 17-8.

Figure 17-18. RTI Compare Up Counter 0 Register (RTICPUC0) [offset = 18h]

rtilnit()



TMS570LC43x 16/32 RISC Flash Microcontroller Technical Reference Manual (Rev. A) 600 / 2208

RTI Control Registers www.ti.com

17.3.5 RTI Free Running Counter 0 Register (RTIFRC0)

The free running counter 0 register holds the current value of free running counter 0. This register is shown in Figure 17-16 and described in Table 17-6.

Figure 17-16. RTI Free Running Counter 0 Register (RTIFRC0) [offset = 10h]

31	16
FRC0	
R/WP-0	

15	0
FRC0	
R/WP-0	

LEGEND: R/W = Read/Write; WP = Write in privileged mode only; -n = value after reset

Table 17-6. RTI Free Running Counter 0 Register (RTIFRC0) Field Descriptions

Bit	Field	Value	Description
31-0	FRC0	0-FFFF FFFFh	Free running counter 0. This registers holds the current value of the free running counter 0. A read of this counter returns the current value of the counter. The counter can be preset by writing (in privileged mode only) to this register. The counter increments then from this written value upwards. **Note: If counters must be preset, they must be disabled in the RTIGCTRL register to ensure consistency between RTIUC0 and RTIFRC0.**

17.3.6 RTI Up Counter 0 Register (RTIUC0)

The up counter 0 register holds the current value of prescale counter. This register is shown in Figure 17-17 and described in Table 17-7.

Figure 17-17. RTI Up Counter 0 Register (RTIUC0) [offset = 14h]

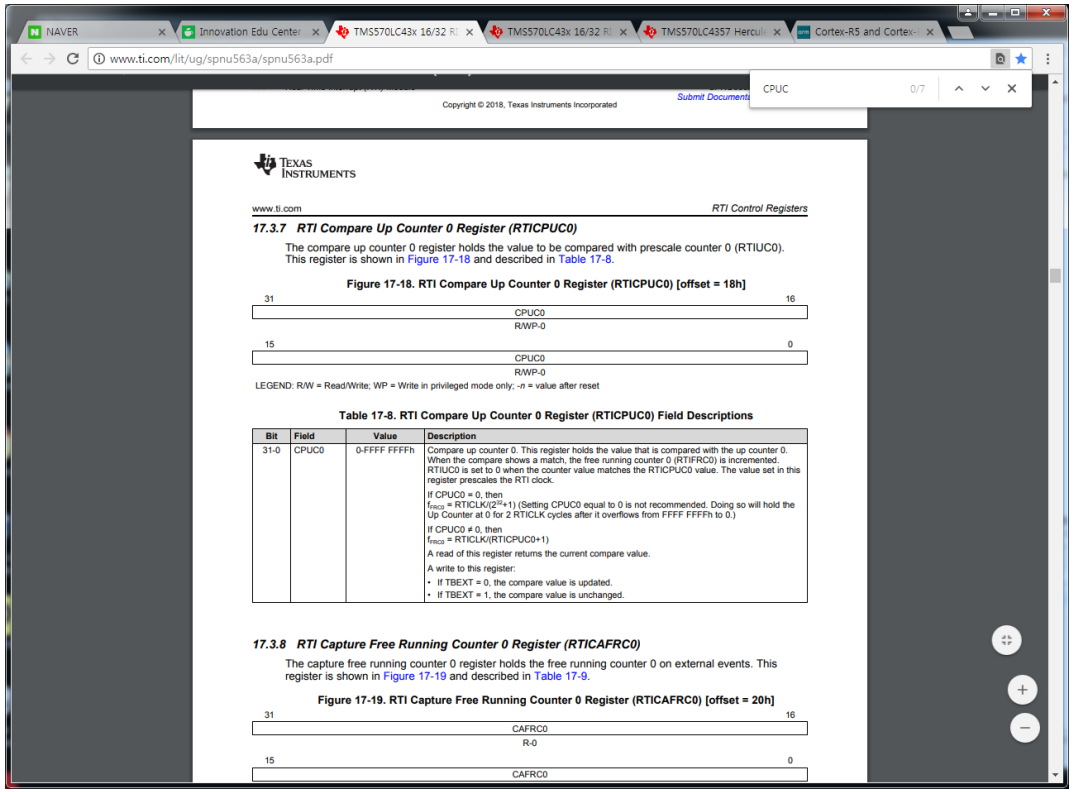
31	16
UC0	
R/WP-0	

15	0
UC0	
R/WP-0	

LEGEND: R/W = Read/Write; WP = Write in privileged mode only; -n = value after reset

Table 17-7. RTI Up Counter 0 Register (RTIUC0) Field Descriptions

Bit	Field	Value	Description
31-0	UC0	0-FFFF FFFFh	Up counter 0. This register 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 (RTIFRC0). This



Copyright © 2018, Texas Instruments Incorporated

RTI Control Registers www.ti.com

17.3.7 RTI Compare Up Counter 0 Register (RTICPUC0)

The compare up counter 0 register holds the value to be compared with prescale counter 0 (RTIUC0). This register is shown in Figure 17-18 and described in Table 17-8.

Figure 17-18. RTI Compare Up Counter 0 Register (RTICPUC0) [offset = 18h]

31	16
CPUC0	
R/WP-0	

15	0
CPUC0	
R/WP-0	

LEGEND: R/W = Read/Write; WP = Write in privileged mode only; -n = value after reset

Table 17-8. RTI Compare Up Counter 0 Register (RTICPUC0) Field Descriptions

Bit	Field	Value	Description
31-0	CPUC0	0-FFFF FFFFh	Compare up counter 0. This register holds the value that is compared with the up counter 0. When the compare shows a match, the free running counter 0 (RTIFRC0) is incremented. RTIUC0 is set to 0 when the counter value matches the RTICPUC0 value. The value set in this register prescales the RTI clock. If CPUC0 = 0, then f_{FRC0} = RTICLK/(2^{32}+1) (Setting CPUC0 equal to 0 is not recommended. Doing so will hold the Up Counter at 0 for 2 RTICLK cycles after it overflows from FFFF FFFFh to 0.) If CPUC0 ≠ 0, then f_{FRC0} = RTICLK/(RTICPUC0+1) A read of this register returns the current compare value. A write to this register: • If TBEXT = 0, the compare value is updated. • If TBEXT = 1, the compare value is unchanged.

17.3.8 RTI Capture Free Running Counter 0 Register (RTICAFRC0)

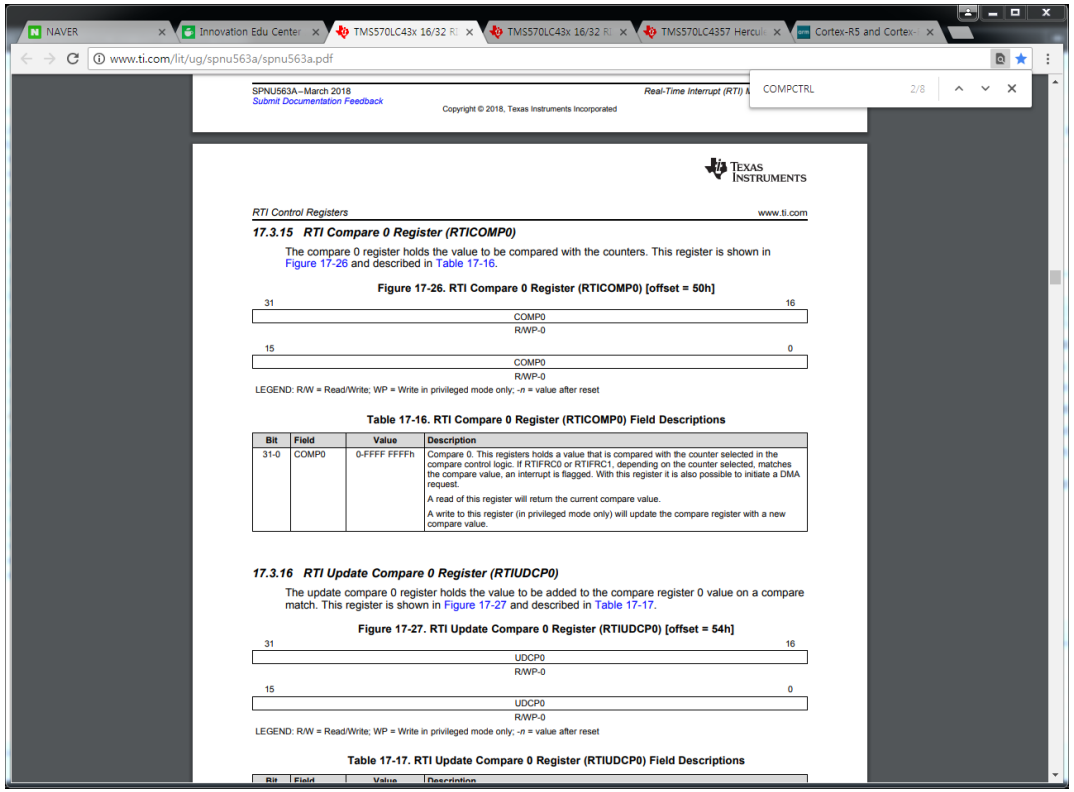
The capture free running counter 0 register holds the free running counter 0 on external events. This register is shown in Figure 17-19 and described in Table 17-9.

Figure 17-19. RTI Capture Free Running Counter 0 Register (RTICAFRC0) [offset = 20h]

31	16
CAFRC0	
R-0	

15	0
CAFRC0	

rtiInit()



NAVER | Innovation Edu Center | TMS570LC43x 16/32 RI | TMS570LC43x 16/32 RI | TMS570LC4357 Hercul | Cortex-R5 and Cortex-

www.ti.com/lit/ug/spnu563a/spnu563a.pdf

COMPCTRL 2/8

17.3.15 RTI Compare 0 Register (RTICOMP0)

The compare 0 register holds the value to be compared with the counters. This register is shown in Figure 17-26 and described in Table 17-16.

Figure 17-26. RTI Compare 0 Register (RTICOMP0) [offset = 50h]

31	16
COMP0	
R/WP-0	

15	0
COMP0	
R/WP-0	

LEGEND: R/W = Read/Write; WP = Write in privileged mode only; -*n* = value after reset

Table 17-16. RTI Compare 0 Register (RTICOMP0) Field Descriptions

Bit	Field	Value	Description
31-0	COMP0	0-FFFF FFFFh	Compare 0. This registers holds a value that is compared with the counter selected in the compare control logic. If RTIFRC0 or RTIFRC1, depending on the counter selected, matches the compare value, an interrupt is flagged. With this register it is also possible to initiate a DMA request. A read of this register will return the current compare value. A write to this register (in privileged mode only) will update the compare register with a new compare value.

17.3.16 RTI Update Compare 0 Register (RTIUDCP0)

The update compare 0 register holds the value to be added to the compare register 0 value on a compare match. This register is shown in Figure 17-27 and described in Table 17-17.

Figure 17-27. RTI Update Compare 0 Register (RTIUDCP0) [offset = 54h]

31	16
UDCP0	
R/WP-0	

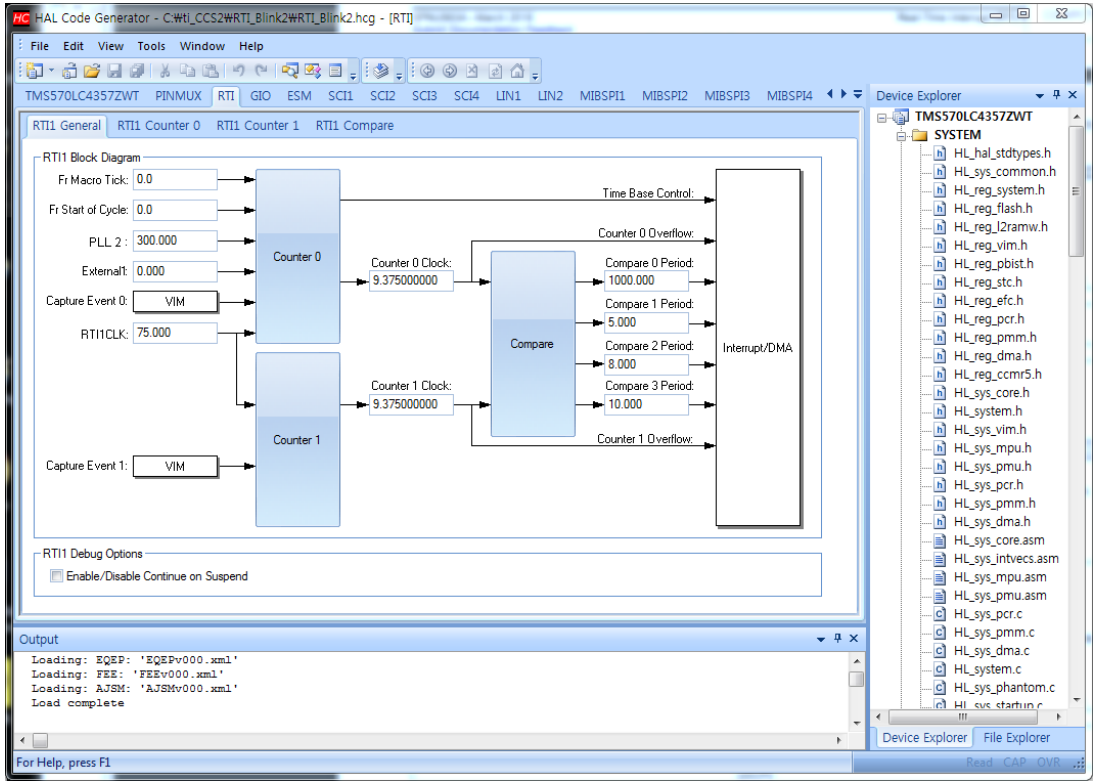
15	0
UDCP0	
R/WP-0	

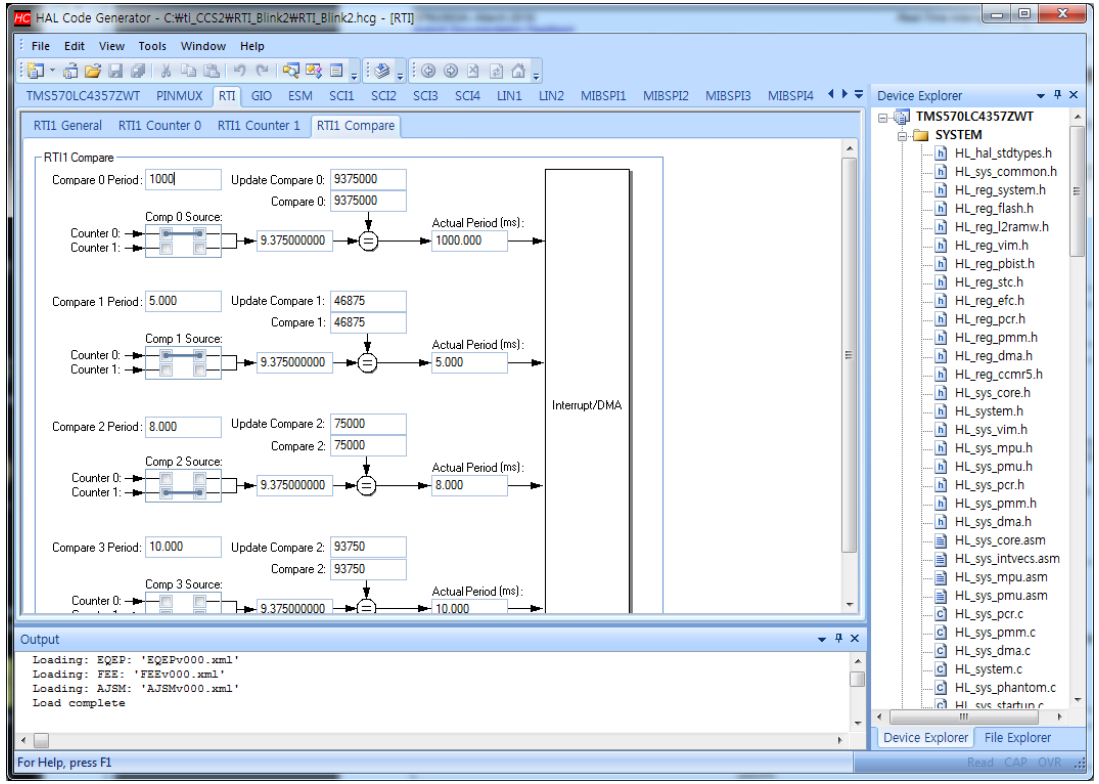
LEGEND: R/W = Read/Write; WP = Write in privileged mode only; -*n* = value after reset

Table 17-17. RTI Update Compare 0 Register (RTIUDCP0) Field Descriptions

Bit	Field	Value	Description

rtiInit()


HAL Code Generator - C:\ti_CCS2\RTI_Blink2\RTI_Blink2.hcg - [RTI]
File Edit View Tools Window Help
TMS570LC4357ZWT PINMUX RTI GIO ESM SCI1 SCI2 SCI3 SCI4 LIN1 LIN2 MIBSPI1 MIBSPI2 MIBSPI3 MIBSPI4
RTI1 General RTI1 Counter 0 RTI1 Counter 1 RTI1 Compare
RTI1 Block Diagram
Fr Macro Tick: 0.0
Fr Start of Cycle: 0.0
PLL 2 : 300.000
ExternalT: 0.000
Capture Event 0: VIM
RTI1CLK: 75.000
Capture Event 1: VIM
Counter 0
Counter 1
Counter 0 Clock: 9.375000000
Counter 1 Clock: 9.375000000
Compare
Time Base Control:
Counter 0 Overflow:
Compare 0 Period: 1000.000
Compare 1 Period: 5.000
Compare 2 Period: 8.000
Compare 3 Period: 10.000
Counter 1 Overflow:
Interrupt/DMA
RTI1 Debug Options
Enable/Disable Continue on Suspend
Output
Loading: EQEP: 'EQEPv000.xml'
Loading: FEE: 'FEEv000.xml'
Loading: AJSM: 'AJSMv000.xml'
Load complete
Device Explorer
TMS570LC4357ZWT
SYSTEM
HL_hal_stdtypes.h
HL_sys_common.h
HL_reg_system.h
HL_reg_flash.h
HL_reg_l2ramw.h
HL_reg_vim.h
HL_reg_pbist.h
HL_reg_stc.h
HL_reg_efc.h
HL_reg_pcr.h
HL_reg_pmm.h
HL_reg_dma.h
HL_reg_ccmr5.h
HL_sys_core.h
HL_system.h
HL_sys_vim.h
HL_sys_mpu.h
HL_sys_pmu.h
HL_sys_pcr.h
HL_sys_pmm.h
HL_sys_dma.h
HL_sys_core.asm
HL_sys_intvecs.asm
HL_sys_mpu.asm
HL_sys_pmu.asm
HL_sys_pcr.c
HL_sys_pmm.c
HL_sys_dma.c
HL_system.c
HL_sys_phantom.c
Device Explorer
File Explorer
For Help, press F1


HAL Code Generator - C:\ti_CCS2\RTI_Blink2\RTI_Blink2.hcg - [RTI]
File Edit View Tools Window Help
TMS570LC4357ZWT PINMUX RTI GIO ESM SCI1 SCI2 SCI3 SCI4 LIN1 LIN2 MIBSPI1 MIBSPI2 MIBSPI3 MIBSPI4
RTI1 General RTI1 Counter 0 RTI1 Counter 1 RTI1 Compare
RTI1 Compare
Compare 0 Period: 1000
Update Compare 0: 9375000
Compare 0: 9375000
Comp 0 Source:
Counter 0:
Counter 1:
9.375000000
Actual Period (ms): 1000.000
Compare 1 Period: 5.000
Update Compare 1: 46875
Compare 1: 46875
Comp 1 Source:
Actual Period (ms): 5.000
Compare 2 Period: 8.000
Update Compare 2: 75000
Compare 2: 75000
Comp 2 Source:
Actual Period (ms): 8.000
Compare 3 Period: 10.000
Update Compare 2: 93750
Compare 2: 93750
Comp 3 Source:
Actual Period (ms): 10.000
Interrupt/DMA
Output
Loading: EQEP: 'EQEPv000.xml'
Loading: FEE: 'FEEv000.xml'
Loading: AJSM: 'AJSMv000.xml'
Load complete
Device Explorer
TMS570LC4357ZWT
SYSTEM
HL_hal_stdtypes.h
HL_sys_common.h
HL_reg_system.h
HL_reg_flash.h
HL_reg_l2ramw.h
HL_reg_vim.h
HL_reg_pbist.h
HL_reg_stc.h
HL_reg_efc.h
HL_reg_pcr.h
HL_reg_pmm.h
HL_reg_dma.h
HL_reg_ccmr5.h
HL_sys_core.h
HL_system.h
HL_sys_vim.h
HL_sys_mpu.h
HL_sys_pmu.h
HL_sys_pcr.h
HL_sys_pmm.h
HL_sys_dma.h
HL_sys_core.asm
HL_sys_intvecs.asm
HL_sys_mpu.asm
HL_sys_pmu.asm
HL_sys_pcr.c
HL_sys_pmm.c
HL_sys_dma.c
HL_system.c
HL_sys_phantom.c
Device Explorer
File Explorer
For Help, press F1

rtiInit()

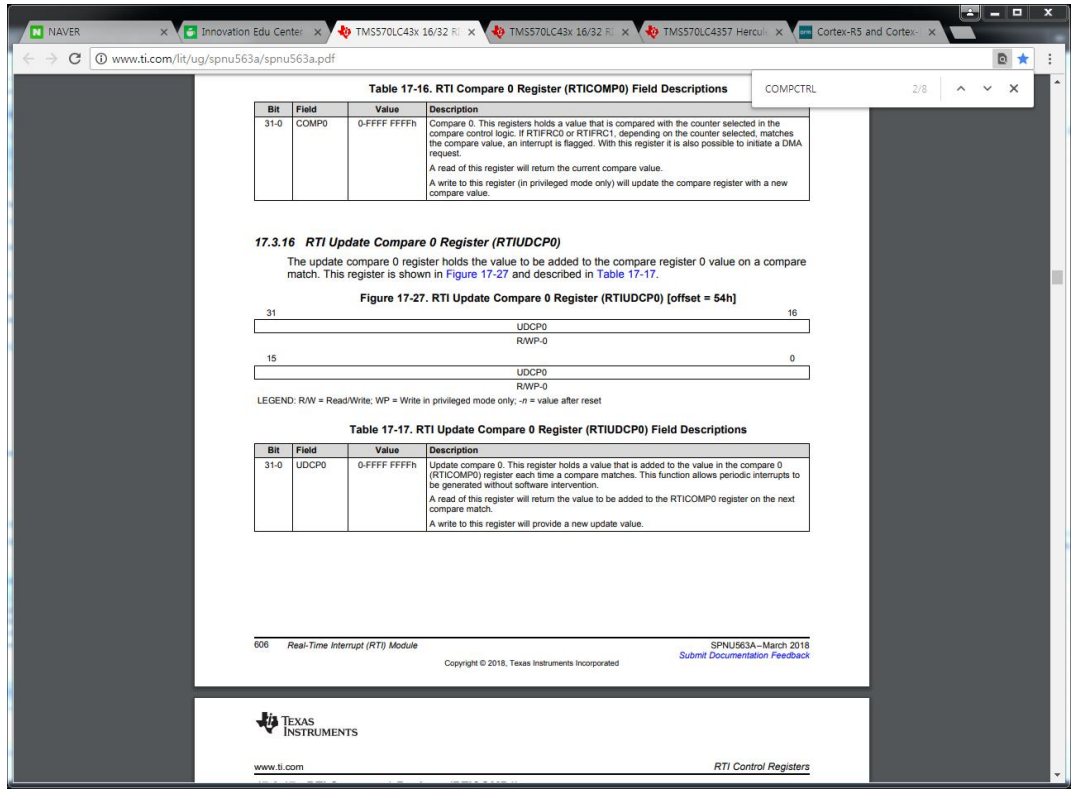


Table 17-16. RTI Compare 0 Register (RTICOMP0) Field Descriptions

Bit	Field	Value	Description
31-0	COMP0	0-FFFF FFFFh	Compare 0. This registers holds a value that is compared with the counter selected in the compare control logic. If RTIFRC0 or RTIFRC1, depending on the counter selected, matches the compare value, an interrupt is flagged. With this register it is also possible to initiate a DMA request. A read of this register will return the current compare value. A write to this register (in privileged mode only) will update the compare register with a new compare value.

17.3.16 RTI Update Compare 0 Register (RTIUDCP0)

The update compare 0 register holds the value to be added to the compare register 0 value on a compare match. This register is shown in Figure 17-27 and described in Table 17-17.

Figure 17-27. RTI Update Compare 0 Register (RTIUDCP0) [offset = 54h]

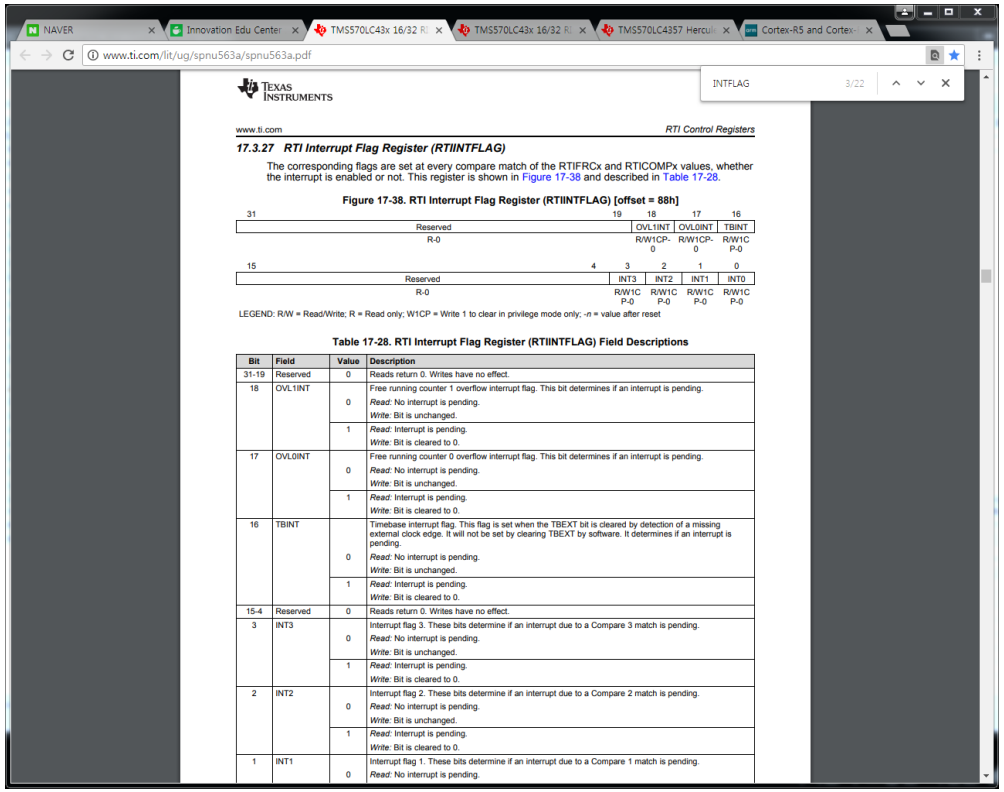
31	16
UDCP0	
R/WP-0	

15	0
UDCP0	
R/WP-0	

LEGEND: R/W = Read/Write; WP = Write in privileged mode only; -n = value after reset

Table 17-17. RTI Update Compare 0 Register (RTIUDCP0) Field Descriptions

Bit	Field	Value	Description
31-0	UDCP0	0-FFFF FFFFh	Update compare 0. This register holds a value that is added to the value in the compare 0 (RTICOMP0) register each time a compare matches. This function allows periodic interrupts to be generated without software intervention. A read of this register will return the value to be added to the RTICOMP0 register on the next compare match. A write to this register will provide a new update value.



17.3.27 RTI Interrupt Flag Register (RTIINTFLAG)

The corresponding flags are set at every compare match of the RTIFRCx and RTICOMPx values, whether the interrupt is enabled or not. This register is shown in Figure 17-38 and described in Table 17-28.

Figure 17-38. RTI Interrupt Flag Register (RTIINTFLAG) [offset = 88h]

31	19	18	17	16
Reserved		OVL1INT	OVL0INT	TBINT
R-0		R/W1CP-0	R/W1CP-0	R/W1CP-0

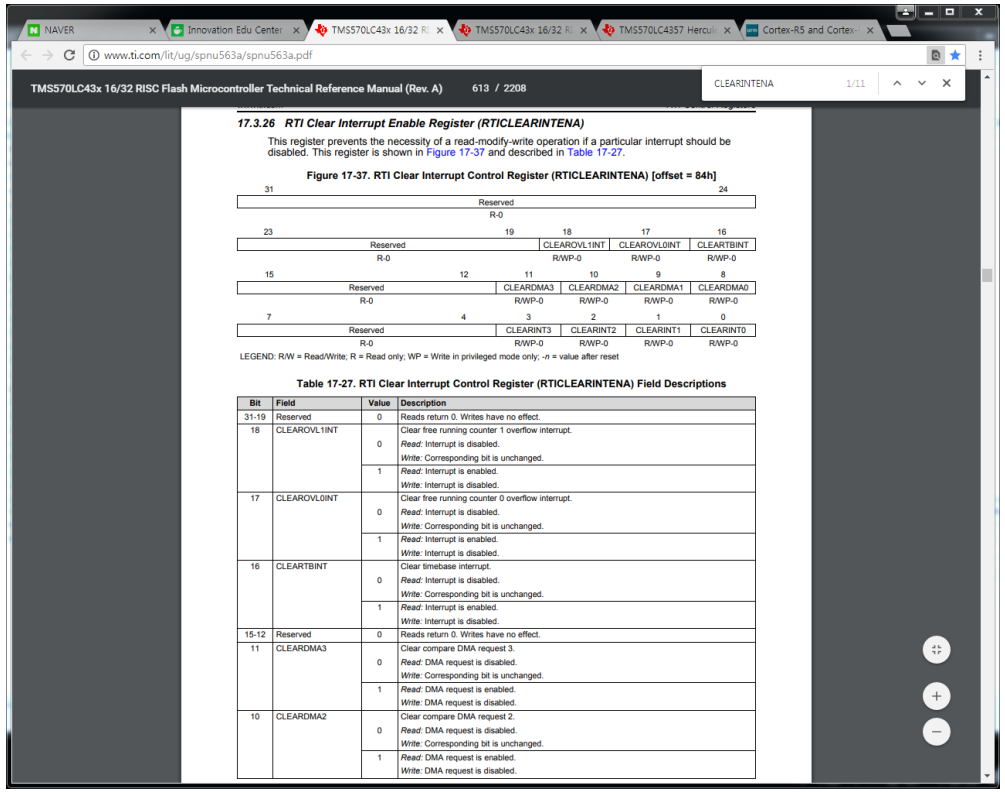
15	4	3	2	1	0
Reserved		INT3	INT2	INT1	INT0
R-0		R/W1CP-0	R/W1CP-0	R/W1CP-0	R/W1CP-0

LEGEND: R/W = Read/Write; R = Read only; W1CP = Write 1 to clear in privilege mode only; -n = value after reset

Table 17-28. RTI Interrupt Flag Register (RTIINTFLAG) Field Descriptions

Bit	Field	Value	Description
31-19	Reserved	0	Reads return 0. Writes have no effect.
18	OVL1INT	0	Free running counter 1 overflow interrupt flag. This bit determines if an interrupt is pending. *Read:* No interrupt is pending. *Write:* Bit is unchanged.
		1	*Read:* Interrupt is pending. *Write:* Bit is cleared to 0.
17	OVL0INT	0	Free running counter 0 overflow interrupt flag. This bit determines if an interrupt is pending. *Read:* No interrupt is pending. *Write:* Bit is unchanged.
		1	*Read:* Interrupt is pending. *Write:* Bit is cleared to 0.
16	TBINT	0	Timebase interrupt flag. This flag is set when the TBEXT bit is cleared by detection of a missing external clock edge. It will not be set by clearing TBEXT by software. It determines if an interrupt is pending. *Read:* No interrupt is pending. *Write:* Bit is unchanged.
		1	*Read:* Interrupt is pending. *Write:* Bit is cleared to 0.
15-4	Reserved	0	Reads return 0. Writes have no effect.
3	INT3	0	Interrupt flag 3. These bits determine if an interrupt due to a Compare 3 match is pending. *Read:* No interrupt is pending. *Write:* Bit is unchanged.
		1	*Read:* Interrupt is pending. *Write:* Bit is cleared to 0.
2	INT2	0	Interrupt flag 2. These bits determine if an interrupt due to a Compare 2 match is pending. *Read:* No interrupt is pending. *Write:* Bit is unchanged.
		1	*Read:* Interrupt is pending. *Write:* Bit is cleared to 0.
1	INT1	0	Interrupt flag 1. These bits determine if an interrupt due to a Compare 1 match is pending. *Read:* No interrupt is pending.

rtiInit()



17.3.26 RTI Clear Interrupt Enable Register (RTICLEARINTENA)

This register prevents the necessity of a read-modify-write operation if a particular interrupt should be disabled. This register is shown in Figure 17-37 and described in Table 17-27.

Figure 17-37. RTI Clear Interrupt Control Register (RTICLEARINTENA) [offset = 84h]

31	24
Reserved	
R-0	

23	19	18	17	16
Reserved		CLEAROVL1INT	CLEAROVL0INT	CLEARTBINT
R-0		R/WP-0	R/WP-0	R/WP-0

15	12	11	10	9	8
Reserved		CLEARDMA3	CLEARDMA2	CLEARDMA1	CLEARDMA0
R-0		R/WP-0	R/WP-0	R/WP-0	R/WP-0

7	4	3	2	1	0
Reserved		CLEARINT3	CLEARINT2	CLEARINT1	CLEARINT0
R-0		R/WP-0	R/WP-0	R/WP-0	R/WP-0

LEGEND: R/W = Read/Write; R = Read only; WP = Write in privileged mode only; -*n* = value after reset

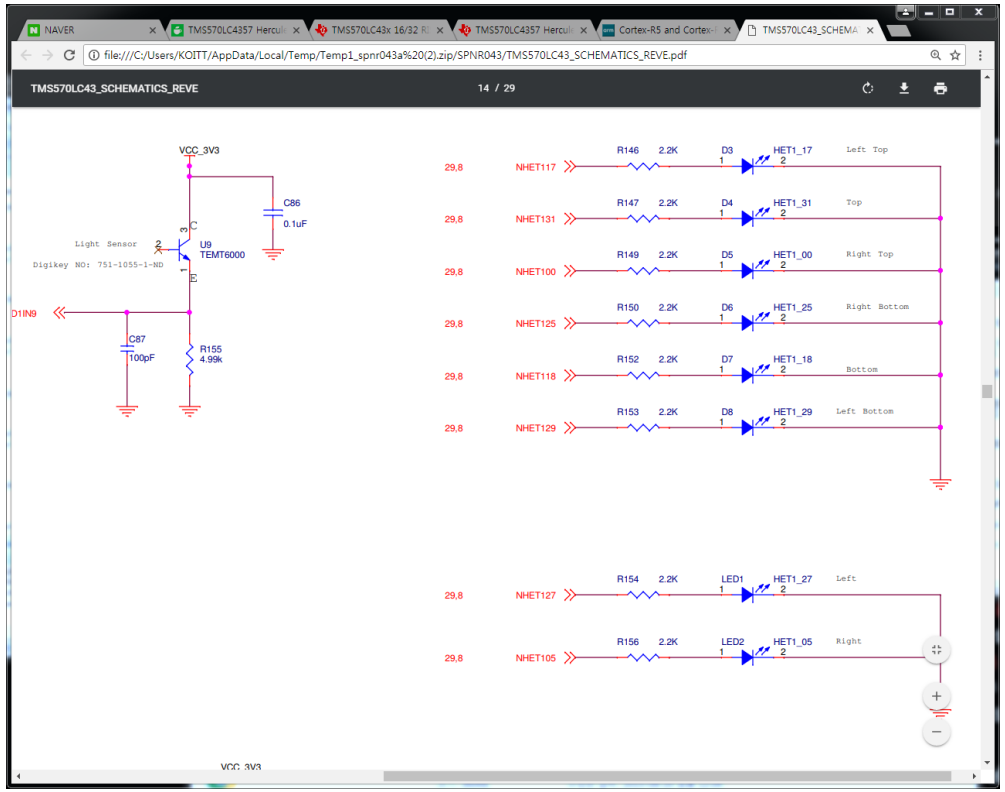
Table 17-27. RTI Clear Interrupt Control Register (RTICLEARINTENA) Field Descriptions

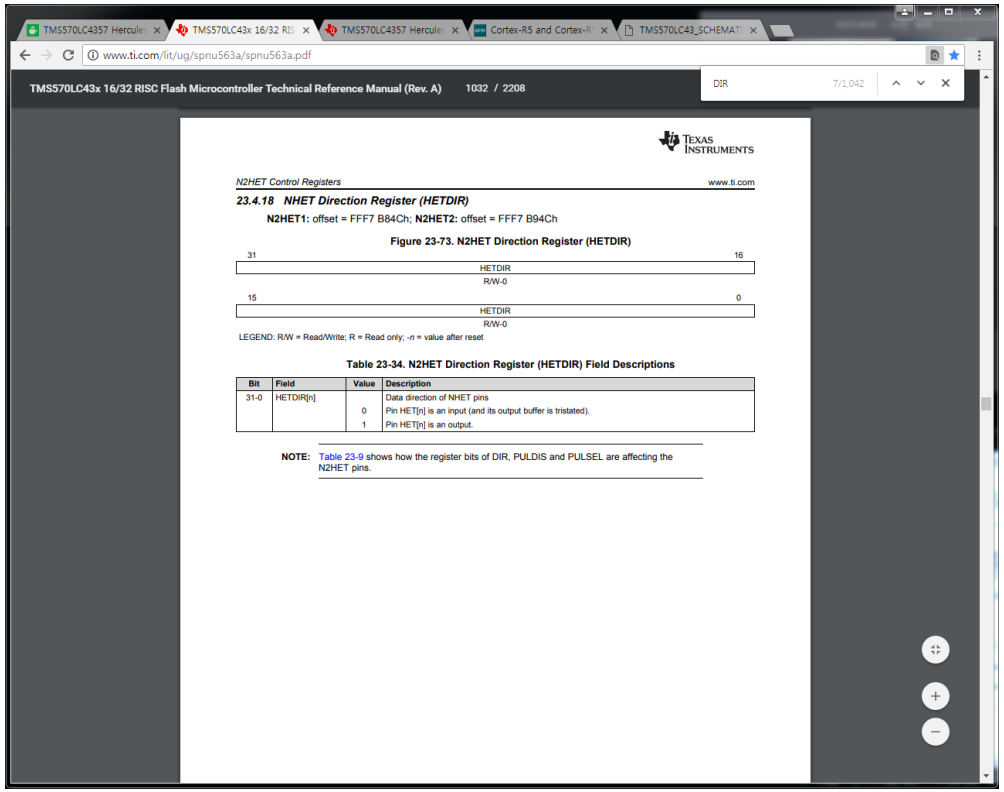
Bit	Field	Value	Description
31-19	Reserved	0	Reads return 0. Writes have no effect.
18	CLEAROVL1INT		Clear free running counter 1 overflow interrupt.
		0	*Read:* Interrupt is disabled. *Write:* Corresponding bit is unchanged.
		1	*Read:* Interrupt is enabled. *Write:* Interrupt is disabled.
17	CLEAROVL0INT		Clear free running counter 0 overflow interrupt.
		0	*Read:* Interrupt is disabled. *Write:* Corresponding bit is unchanged.
		1	*Read:* Interrupt is enabled. *Write:* Interrupt is disabled.
16	CLEARTBINT		Clear timebase interrupt.
		0	*Read:* Interrupt is disabled. *Write:* Corresponding bit is unchanged.
		1	*Read:* Interrupt is enabled. *Write:* Interrupt is disabled.
15-12	Reserved	0	Reads return 0. Writes have no effect.
11	CLEARDMA3		Clear compare DMA request 3.
		0	*Read:* DMA request is disabled. *Write:* Corresponding bit is unchanged.
		1	*Read:* DMA request is enabled. *Write:* DMA request is disabled.
10	CLEARDMA2		Clear compare DMA request 2.
		0	*Read:* DMA request is disabled. *Write:* Corresponding bit is unchanged.
		1	*Read:* DMA request is enabled. *Write:* DMA request is disabled.

gioSetDirection

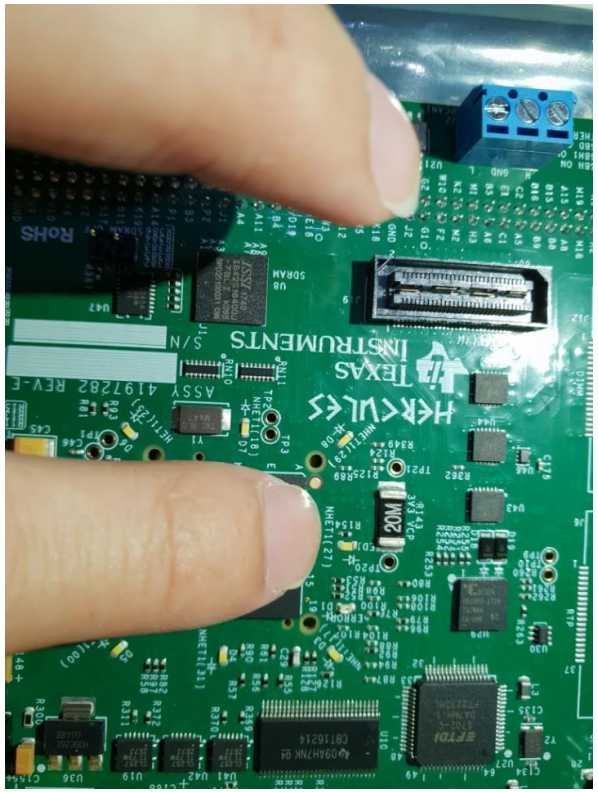
```
gioSetDirection(hetPORT1, 0xFFFFFFFF);

        #define hetPORT1 ((gioPORT_t *)0xFFF7B84CU)

        void gioSetDirection(gioPORT_t *port, uint32 dir)
        {
                port->DIR = dir;
        }
```

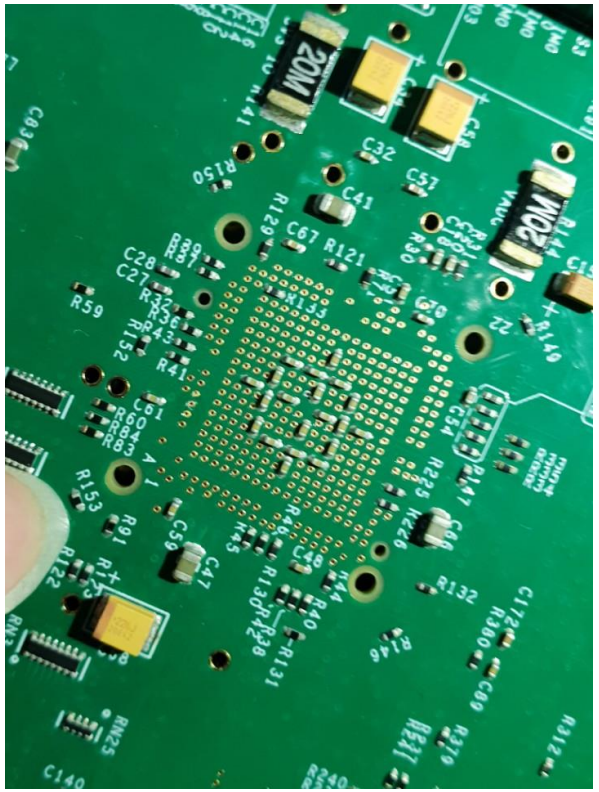
gioSetDirection





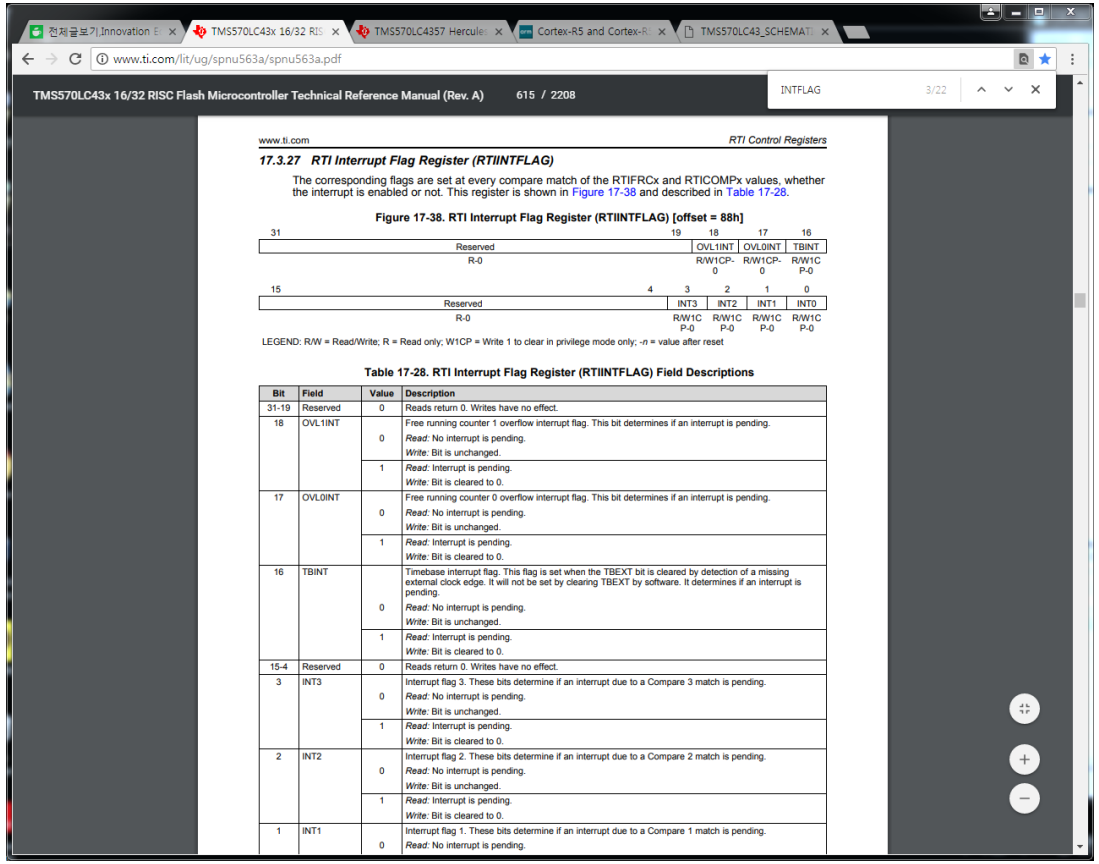
gioSetDirection





RTI

```
rtiEnableNotification(rtiREG1, rtiNOTIFICATION_COMPARE0)

        #define rtiNOTIFICATION_COMPARE0 1U

        rtiREG->INTFLAG = notification;
        rtiREG->SETINTENA   = notification;
```



www.ti.com/lit/ug/spnu563a/spnu563a.pdf

TMS570LC43x 16/32 RISC Flash Microcontroller Technical Reference Manual (Rev. A) 615 / 2208

INTFLAG 3/22

www.ti.com RTI Control Registers

17.3.27 RTI Interrupt Flag Register (RTIINTFLAG)

The corresponding flags are set at every compare match of the RTIFRCx and RTICOMPx values, whether the interrupt is enabled or not. This register is shown in Figure 17-38 and described in Table 17-28.

Figure 17-38. RTI Interrupt Flag Register (RTIINTFLAG) [offset = 88h]

31	19	18	17	16
Reserved		OVL1INT	OVL0INT	TBINT
R-0		R/W1CP-0	R/W1CP-0	R/W1CP-0

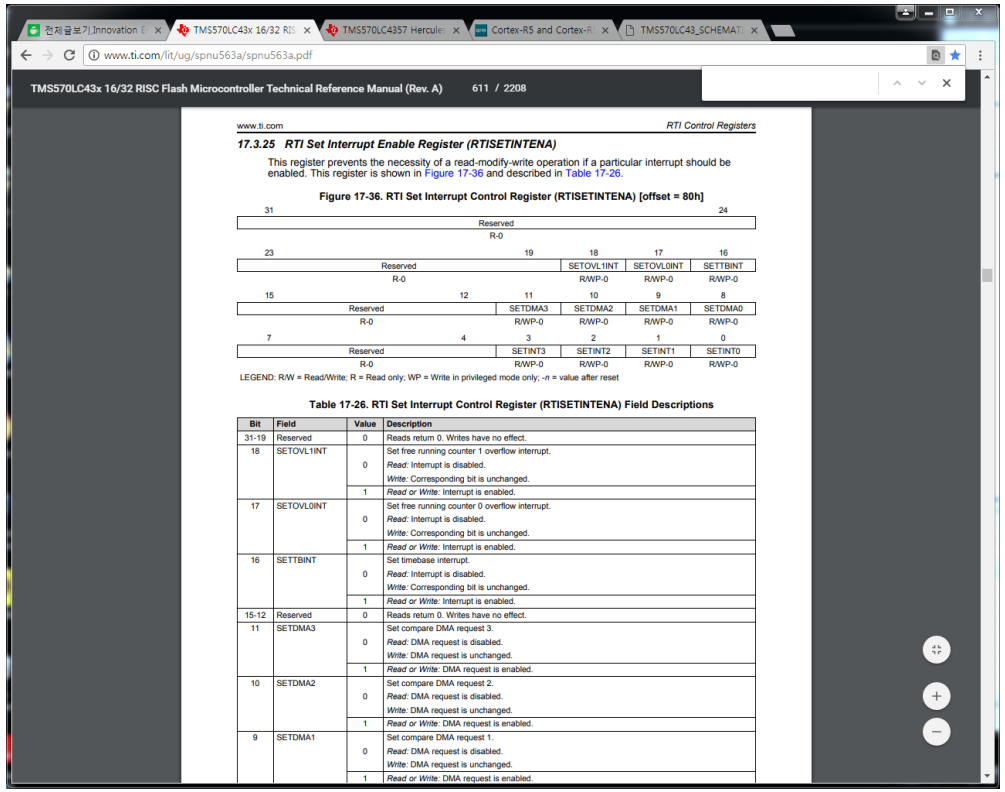
15	4	3	2	1	0
Reserved		INT3	INT2	INT1	INT0
R-0		R/W1CP-0	R/W1CP-0	R/W1CP-0	R/W1CP-0

LEGEND: R/W = Read/Write; R = Read only; W1CP = Write 1 to clear in privilege mode only; -n = value after reset

Table 17-28. RTI Interrupt Flag Register (RTIINTFLAG) Field Descriptions

Bit	Field	Value	Description
31-19	Reserved	0	Reads return 0. Writes have no effect.
18	OVL1INT		Free running counter 1 overflow interrupt flag. This bit determines if an interrupt is pending.
		0	*Read:* No interrupt is pending. *Write:* Bit is unchanged.
		1	*Read:* Interrupt is pending. *Write:* Bit is cleared to 0.
17	OVL0INT		Free running counter 0 overflow interrupt flag. This bit determines if an interrupt is pending.
		0	*Read:* No interrupt is pending. *Write:* Bit is unchanged.
		1	*Read:* Interrupt is pending. *Write:* Bit is cleared to 0.
16	TBINT		Timebase interrupt flag. This flag is set when the TBEXT bit is cleared by detection of a missing external clock edge. It will not be set by clearing TBEXT by software. It determines if an interrupt is pending.
		0	*Read:* No interrupt is pending. *Write:* Bit is unchanged.
		1	*Read:* Interrupt is pending. *Write:* Bit is cleared to 0.
15-4	Reserved	0	Reads return 0. Writes have no effect.
3	INT3		Interrupt flag 3. These bits determine if an interrupt due to a Compare 3 match is pending.
		0	*Read:* No interrupt is pending. *Write:* Bit is unchanged.
		1	*Read:* Interrupt is pending. *Write:* Bit is cleared to 0.
2	INT2		Interrupt flag 2. These bits determine if an interrupt due to a Compare 2 match is pending.
		0	*Read:* No interrupt is pending. *Write:* Bit is unchanged.
		1	*Read:* Interrupt is pending. *Write:* Bit is cleared to 0.
1	INT1		Interrupt flag 1. These bits determine if an interrupt due to a Compare 1 match is pending.
		0	*Read:* No interrupt is pending.

RTI



www.ti.com/lit/ug/spnu563a/spnu563a.pdf

TMS570LC43x 16/32 RISC Flash Microcontroller Technical Reference Manual (Rev. A) 611 / 2208

17.3.25 RTI Set Interrupt Enable Register (RTISETINTENA)

This register prevents the necessity of a read-modify-write operation if a particular interrupt should be enabled. This register is shown in Figure 17-36 and described in Table 17-26.

Figure 17-36. RTI Set Interrupt Control Register (RTISETINTENA) [offset = 80h]

31–24	23–19	18	17	16
Reserved	Reserved	SETOVL1INT	SETOVL0INT	SETTBINT
R-0	R-0	R/WP-0	R/WP-0	R/WP-0

15–12	11	10	9	8
Reserved	SETDMA3	SETDMA2	SETDMA1	SETDMA0
R-0	R/WP-0	R/WP-0	R/WP-0	R/WP-0

7–4	3	2	1	0
Reserved	SETINT3	SETINT2	SETINT1	SETINT0
R-0	R/WP-0	R/WP-0	R/WP-0	R/WP-0

LEGEND: R/W = Read/Write; R = Read only; WP = Write in privileged mode only; -n = value after reset

Table 17-26. RTI Set Interrupt Control Register (RTISETINTENA) Field Descriptions

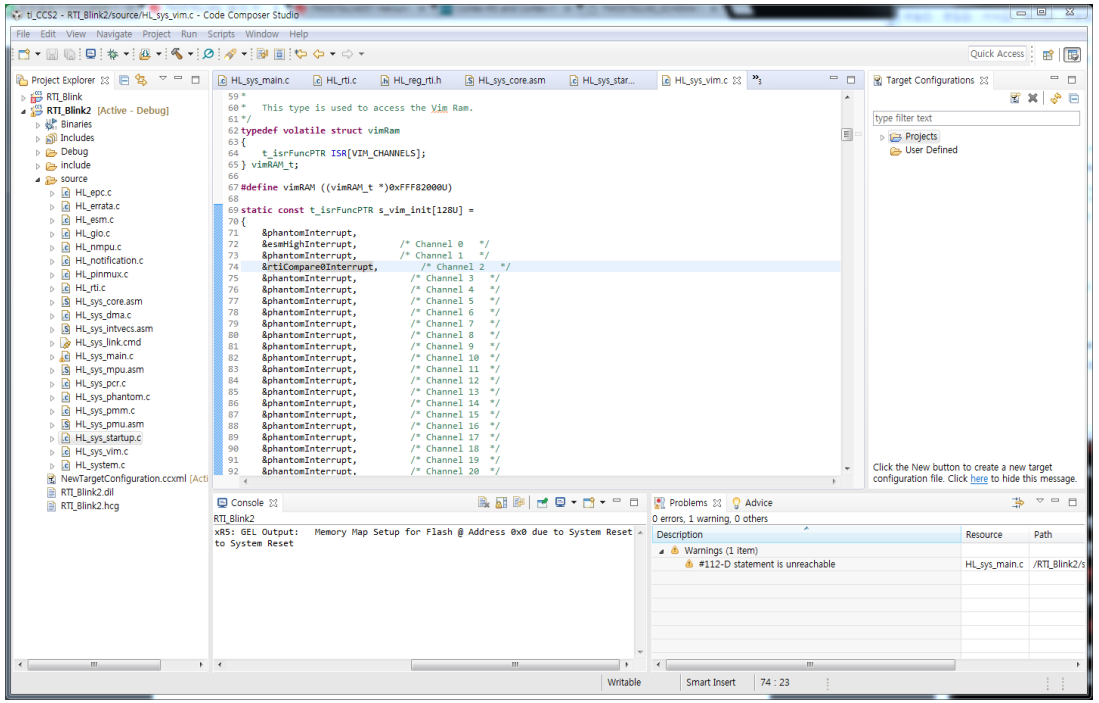
Bit	Field	Value	Description
31-19	Reserved	0	Reads return 0. Writes have no effect.
18	SETOVL1INT	0	Set free running counter 1 overflow interrupt. *Read:* Interrupt is disabled. *Write:* Corresponding bit is unchanged.
		1	*Read or Write:* Interrupt is enabled.
17	SETOVL0INT	0	Set free running counter 0 overflow interrupt. *Read:* Interrupt is disabled. *Write:* Corresponding bit is unchanged.
		1	*Read or Write:* Interrupt is enabled.
16	SETTBINT	0	Set timebase interrupt. *Read:* Interrupt is disabled. *Write:* Corresponding bit is unchanged.
		1	*Read or Write:* Interrupt is enabled.
15-12	Reserved	0	Reads return 0. Writes have no effect.
11	SETDMA3	0	Set compare DMA request 3. *Read:* DMA request is disabled. *Write:* DMA request is unchanged.
		1	*Read or Write:* DMA request is enabled.
10	SETDMA2	0	Set compare DMA request 2. *Read:* DMA request is disabled. *Write:* DMA request is unchanged.
		1	*Read or Write:* DMA request is enabled.
9	SETDMA1	0	Set compare DMA request 1. *Read:* DMA request is disabled. *Write:* DMA request is unchanged.
		1	*Read or Write:* DMA request is enabled.

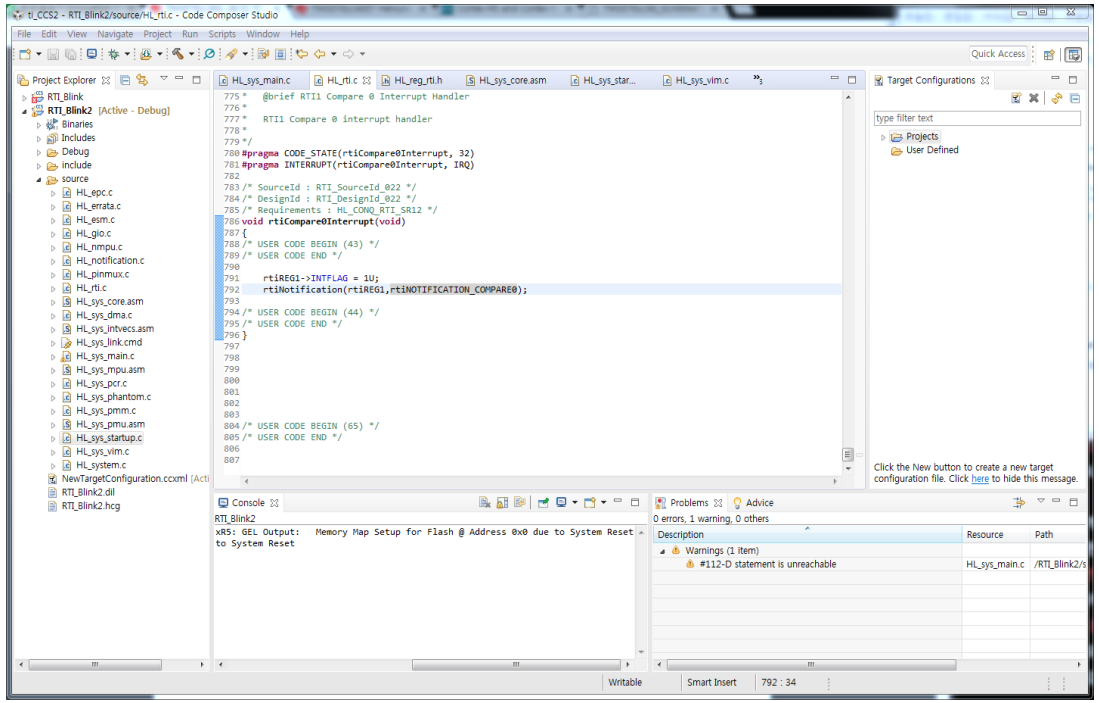
IRQ

```
_enable_IRQ_interrupt_();
```

```
cpsie i
bx    lr
```

IRQ


tj_CCS2 - RTI_Blink2/source/HL_sys_vim.c - Code Composer Studio
File Edit View Navigate Project Run Scripts Window Help
Quick Access
Project Explorer
RTI_Blink
RTI_Blink2 [Active - Debug]
Binaries
Includes
Debug
include
source
HL_epc.c
HL_errata.c
HL_esm.c
HL_gio.c
HL_nmpu.c
HL_notification.c
HL_pinmux.c
HL_rti.c
HL_sys_core.asm
HL_sys_dma.c
HL_sys_intvecs.asm
HL_sys_link.cmd
HL_sys_main.c
HL_sys_mpu.asm
HL_sys_pcr.c
HL_sys_phantom.c
HL_sys_pmm.c
HL_sys_pmu.asm
HL_sys_startup.c
HL_sys_vim.c
HL_system.c
NewTargetConfiguration.ccxml [Acti
RTI_Blink2.dil
RTI_Blink2.hcg
HL_sys_main.c
HL_rti.c
HL_reg_rti.h
HL_sys_core.asm
HL_sys_star...
HL_sys_vim.c
59 *
60 * This type is used to access the Vim Ram.
61 */
62typedef volatile struct vimRam
63{
64 t_isrFuncPTR ISR[VIM_CHANNELS];
65} vimRAM_t;
66
67#define vimRAM ((vimRAM_t *)0xFFF82000U)
68
69static const t_isrFuncPTR s_vim_init[128U] =
70{
71 &phantomInterrupt,
72 &esmHighInterrupt, /* Channel 0 */
73 &phantomInterrupt, /* Channel 1 */
74 &rtiCompare0Interrupt, /* Channel 2 */
75 &phantomInterrupt, /* Channel 3 */
76 &phantomInterrupt, /* Channel 4 */
77 &phantomInterrupt, /* Channel 5 */
78 &phantomInterrupt, /* Channel 6 */
79 &phantomInterrupt, /* Channel 7 */
80 &phantomInterrupt, /* Channel 8 */
81 &phantomInterrupt, /* Channel 9 */
82 &phantomInterrupt, /* Channel 10 */
83 &phantomInterrupt, /* Channel 11 */
84 &phantomInterrupt, /* Channel 12 */
85 &phantomInterrupt, /* Channel 13 */
86 &phantomInterrupt, /* Channel 14 */
87 &phantomInterrupt, /* Channel 15 */
88 &phantomInterrupt, /* Channel 16 */
89 &phantomInterrupt, /* Channel 17 */
90 &phantomInterrupt, /* Channel 18 */
91 &phantomInterrupt, /* Channel 19 */
92 &phantomInterrupt, /* Channel 20 */
Target Configurations
type filter text
Projects
User Defined
Click the New button to create a new target configuration file. Click here to hide this message.
Console
RTI_Blink2
xR5: GEL Output: Memory Map Setup for Flash @ Address 0x0 due to System Reset to System Reset
Problems
Advice
0 errors, 1 warning, 0 others
Description
Resource
Path
Warnings (1 item)
#112-D statement is unreachable
HL_sys_main.c
/RTI_Blink2/s
Writable
Smart Insert
74 : 23


tj_CCS2 - RTI_Blink2/source/HL_rti.c - Code Composer Studio
File Edit View Navigate Project Run Scripts Window Help
Quick Access
Project Explorer
RTI_Blink
RTI_Blink2 [Active - Debug]
Binaries
Includes
Debug
include
source
HL_epc.c
HL_errata.c
HL_esm.c
HL_gio.c
HL_nmpu.c
HL_notification.c
HL_pinmux.c
HL_rti.c
HL_sys_core.asm
HL_sys_dma.c
HL_sys_intvecs.asm
HL_sys_link.cmd
HL_sys_main.c
HL_sys_mpu.asm
HL_sys_pcr.c
HL_sys_phantom.c
HL_sys_pmm.c
HL_sys_pmu.asm
HL_sys_startup.c
HL_sys_vim.c
HL_system.c
NewTargetConfiguration.ccxml [Acti
RTI_Blink2.dil
RTI_Blink2.hcg
HL_sys_main.c
HL_rti.c
HL_reg_rti.h
HL_sys_core.asm
HL_sys_star...
HL_sys_vim.c
775 * @brief RTI1 Compare 0 Interrupt Handler
776 *
777 * RTI1 Compare 0 interrupt handler
778 *
779 */
780#pragma CODE_STATE(rtiCompare0Interrupt, 32)
781#pragma INTERRUPT(rtiCompare0Interrupt, IRQ)
782
783/* SourceId : RTI_SourceId_022 */
784/* DesignId : RTI_DesignId_022 */
785/* Requirements : HL_CONQ_RTI_SR12 */
786void rtiCompare0Interrupt(void)
787{
788/* USER CODE BEGIN (43) */
789/* USER CODE END */
790
791 rtiREG1->INTFLAG = 1U;
792 rtiNotification(rtiREG1,rtiNOTIFICATION_COMPARE0);
793
794/* USER CODE BEGIN (44) */
795/* USER CODE END */
796}
797
798
799
800
801
802
803
804/* USER CODE BEGIN (65) */
805/* USER CODE END */
806
807
Target Configurations
type filter text
Projects
User Defined
Click the New button to create a new target configuration file. Click here to hide this message.
Console
RTI_Blink2
xR5: GEL Output: Memory Map Setup for Flash @ Address 0x0 due to System Reset to System Reset
Problems
Advice
0 errors, 1 warning, 0 others
Description
Resource
Path
Warnings (1 item)
#112-D statement is unreachable
HL_sys_main.c
/RTI_Blink2/s
Writable
Smart Insert
792 : 34

문제

17 31 0

27 5

29 18 25

A A 0 6 0 0 2 1

1010 1010 0000 0110 0000 0000 0010 0001

문제

1. N2HET 을 활용해 LED 8 개 전부 키기

```
gioSetPort(hetPORT1, gioGetPort(hetPORT1) ^ 0xAA020021);
```

2. LED 를 뱀처럼 1 개 키고 그 다음거 키고(2 개 켜짐) 그 다음거 키고 (3개 켜짐) 마지막에 8 개 키고 다시 반복

```
if(count == 8)
{
    gioSetPort(hetPORT1, 0x00000000);
    count = 0;
    return;
}
gioSetPort(hetPORT1, gioGetPort(hetPORT1) ^ GIO_het[count++]);
```

문제

3. LED 를 순서대로 1 개씩 키기(다음거 킬때 기존에 켜져 있던것이 꺼져야함)

```
if(count == 8)
        count = 0;
gioSetPort(hetPORT1, GIO_het[count++]);
```

4. 랜덤으로 LED 가 켜지게 함

```
srand(time(NULL));

count = rand()%8;
gioSetPort(hetPORT1, GIO_het[count]);
```