RENESAS Tool News

RENESAS TOOL NEWS on May 16, 2015: 150516/tn1

Note on Using the CS+ Code Generator for RX

When using the CS+ Code_Generator for RX, take note of the problems on the following points that are described in this note.

1. Code Generated for the Clock Generation Circuit (PLL Circuit Operation)
   Applicable products: RX111 and RX113 groups
2. Bus Settings
   Applicable products: RX64M and RX71M groups
3. 16-bit Timer Pulse Unit (TPUa) and Multifunction Timer Pulse Unit 3 (MTU3a)
   Applicable products: RX64M and RX71M groups
4. 12-bit A/D Converter (S12ADC)
   Applicable products: RX64M and RX71M groups
5. 12-bit D/A Converter (R12DA)
   Applicable products: RX64M and RX71M groups

1. Code Generated for the Clock Generation Circuit (PLL Circuit Operation)

1.1 Product Concerned

V1.03.00 and later versions of the CS+ Code Generator for RX

1.2 MCUs Involved

RX family: RX111 and RX113 groups

1.3 Description

When "Operation" is selected under "PLL circuit setting" on the "Clock Generator" page, the generated code has an error.

1.4 Workaround

Modify the output code of void R_CGC_Create(void) in the way shown below. The function is in the r_cg_cgc.c file. This modification is required every time code is generated.

Before modification:

```
----------------------------------------------------------------------
void R_CGC_Create(void)
```

```
{
.............
   /* Set PLL circuit */
   SYSTEM.PLLCR2.BIT.PLLEN = 0U;          <- faulty sequence
   SYSTEM.PLLCR.WORD = _0001_CGC_PLL_FREQ_DIV_2 |
                   _0B00_CGC_PLL_FREQ_MUL_6;
.............
}
--------------------------------------------------------------
```

After modification:

```
--------------------------------------------------------------
void R_CGC_Create(void)
{
.............
   /* Set PLL circuit */
   SYSTEM.PLLCR.WORD = _0001_CGC_PLL_FREQ_DIV_2 |
                   _0B00_CGC_PLL_FREQ_MUL_6;
   SYSTEM.PLLCR2.BIT.PLLEN = 0U;   <- correct sequence
                                  That is, modify the order of
                                  processing.
.............
}
--------------------------------------------------------------
```

1.5 Schedule for Fixing the Problem

This problem will be fixed in the next version.

2. Bus Settings

2.1 Products Concerned

V1.05.00 and later versions of the CS+ Code Generator for RX

2.2 MCUs Involved

RX family: RX64M and RX71M groups

2.3 Description

An error is generated in the address output pins "A16-A23" when the following address output pin settings are made for the bus.

(1) "P71" is selected as the CS1#output pin after checking "Use CS1" with the bus operation setting "Used".

(2) After selecting "Normal mode" for TPU1 in the 16-bit timer pulse unit settings, "P14" is enabled and set as the TCLKA pin.

(3) General registers TGRA1 and TGRB1 of TPU1 are set as "Output compare register", after which pins TIOCA1 and TIOCB1 are set to "output

disabled".

(4) The address output pin settings for the bus are made.

2.4. Workaround

There is no way to prevent this problem.
"A16-A23" cannot be set as address output pins.

2.5. Schedule for Fixing the Problem

This problem will be fixed in the next version.

3. 16-bit Timer Pulse Unit (TPUa) and Multifunction Timer Pulse Unit 3 (MTU3a)

3.1 Products Concerned

V1.05.00 and later versions of the CS+ Code Generator for RX

3.2 MCUs Involved

RX family: RX64M and RX71M groups

3.3 Description

A selection error is generated for the TIOCA1 pin of TPU1 of the 16-bit timer pulse unit when the following settings for the unit are made.

(1) TPU1 of the 16-bit timer pulse unit is set to "Phase counting mode 1", a check mark is placed against "TCLKA pin", and "P14" is selected as the TCLKA pin.

(2) The TIOCA1 and TIOCB1 pins of TPU1 are set to "Input capture at TPU0.TGRn input capture/compare match".

(3) MTU0 of the Multifunction Timer Pulse Unit 3 is set to "Normal mode" and a check mark is placed against "MTCLKA pin".

3.4 Workaround

There is no way to prevent this problem.
The TIOCA1 pin cannot be used with TPU1 of the 16-bit timer pulse unit.

3.5. Schedule for Fixing the Problem

This problem will be fixed in the next version.

4. 12-bit A/D Converter (S12ADC)

4.1 Products Concerned

V1.05.00 and later versions of the CS+ Code Generator for RX

4.2 MCUs Involved

RX family: RX64M and RX71M groups

4.3 Description

When the 12-bit A/D converter (S12ADC) is used in the group scan mode, an A/D conversion end interrupt or group B A/D conversion end interrupt will be generated immediately after the scan has started, if further scanning is started by calling the function void R_S12ADn_Start(void) after the function void R_S12ADn_Stop(void) has been executed.

4.4 Workaround

Add settings for the following registers before calls of the function void R_S12ADn_Stop(void). Refer to "Added register settings" for details of the settings.

- ADGSPCR register
- ADSTRGR register
- ADCSR register

Add settings for the following register if you are using "scan end event" with the event link controller.

- ELSRn register

Initialize the registers after executing the function void R_S12ADn_Create(void) when restarting A/D conversion after a call of the function void R_S12ADx_Stop(void).

Added register settings:

```
--------------------------------------------------------------
/* Cancel priority group */
if (S12ADn.ADGSPCR.PGS == 1) /* Does group A have priority? */
{
   S12ADn.ADGSPCR.PGS = 0;  /* Group A does not have priority. */
}

/* Initialize the trigger and disable interrupts. */
S12ADn.ADSTGR.TRSA = 0x3FU;
S12ADn.ADSCR.ADIE = 0;
if(S12ADn.ADSCR.ADCS == 1;   /* Group scanning? */
{
   S12ADn.ADSTGR.TRSB = 0x3FU;
   S12ADn.ADSCR.GBADIE = 0;
}

/* Clear the source for the ELC. */
ELC.ELSRn.ELS = 0x00U;
--------------------------------------------------------------
```

4.5 Schedule for Fixing the Problem

This problem will be fixed in the next version.

5. 12-bit D/A Converter (R12DA)

5.1 Products Concerned

V1.05.00 and later versions of the CS+ Code Generator for RX

5.2 MCUs Involved

RX family: RX64M and RX71M groups

5.3 Description

The code output for the function void R_R12DAx_Start(void) has an error, so the output amplifier may not operate correctly in use with the 12-bit D/A converter (R12DA).

5.4 Workaround

Modify the code output for void R_R12DA0_Start(void) in the way shown below. The function is in the r_cg_r12da.c file. This modification is required every time code is generated.

Before modification:

```
-------------------------------------------------------
DA.DADR0 = 0x0000U;
DA.DACR.BIT.DAE = 0U;
DA.DACR.BIT.DAOE0 = 1U;
DA.DAAMPCR.BIT.DAAMP0 = 1U;    <- faulty sequence
-------------------------------------------------------
```

After modification:

```
-------------------------------------------------------
DA.DADR0 = 0x0000U;
DA.DAAMPCR.BIT.DAAMP0 = 1U;    <- correct sequence
                                  That is, modify the order of
                                  processing.
DA.DACR.BIT.DAE = 0U;
DA.DACR.BIT.DAOE0 = 1U;
-------------------------------------------------------
```

5.5 Schedule for Fixing the Problem

This problem will be fixed in the next version.

[Disclaimer]

The past news contents have been based on information at the time of publication. Now changed or invalid information may be included. The URLs in the Tool News also may be subject to change or become invalid without prior notice.