R-Car S4 series
Modifying the LPDDR4X SDRAM Initialization Code (Rev.1.05)

Introduction
This application note describes the R-Car S4 series LPDDR4X SDRAM initialization program code we provide. Users may need to modify the code to suit their own hardware configurations.

Target Device
R-Car S4 Series

Target Source Code Revision
rev.0.50

Note: The revision of the LPDDR4X SDRAM initialization code is defined as “R_DDR_VERSION” in “boot_init_dram_regdef.h”.

## Contents

1. DRAM Initialization Software ................................................................. 3
   1.1 List of Source Files ............................................................................. 3

2. Customizing boot_init_dram_config.c/ boot_init_dram_config.h .............. 4
   2.1 Customizing boot_init_dram_config.c/ boot_init_dram_config.h ............ 5
      2.1.1 Setting of board configuration number ........................................... 5
      2.1.2 Definition of R-Car S4 16bit LPDDR4X SDRAM I/F CH number .......... 6
      2.1.3 Customization of boardcnf[] structure (1) ......................................... 7
      2.1.4 Customization of boardcnf[] structure (2) ......................................... 9
      2.1.5 Customization of boardcnf[] structure (3) ....................................... 10
      2.1.6 Customization of boardcnf[] structure (4) ....................................... 11
      2.1.7 Customization of boardcnf[] structure (5) ....................................... 13
      2.2 LPDDR4X SDRAM refresh rate adjusting ........................................... 15
      2.3 LPDDR4X SDRAM temperature derating configuration ...................... 16
      2.4 Periodic Write DQ training interval configuration ............................... 17
      2.5 Customizing void boardcnf_get_brd_clk() function .............................. 18
      2.6 Customizing void boardcnf_get_ddr_mbps() function ............................ 19

Revision History ....................................................................................... 21
1. DRAM Initialization Software

1.1 List of Source Files

The source code consists of two sets of files. One set is not to be modified by the user, and the other requires modification by users to suit their own hardware.

<table>
<thead>
<tr>
<th>File Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>boot_init_dram.c (*1)</td>
</tr>
<tr>
<td>boot_init_dram_regdef.h</td>
</tr>
<tr>
<td>ddr_regdef.h</td>
</tr>
<tr>
<td>init_dram_tbl_s4.h</td>
</tr>
</tbody>
</table>

(*1) If DRAM back-up, which is sometime called ‘Suspend to RAM’ in Linux world, is not needed. DRAM backup mode configuration (#define DDR_BACKUPMODE) must comment out.

<table>
<thead>
<tr>
<th>File Name</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>boot_init_dram_config.c</td>
<td>The functions to set the configuration specific to the hardware, including the DRAM capacity and hardware connections, are defined in this file.</td>
</tr>
<tr>
<td></td>
<td>r_boardcnf_get_brd_type()</td>
</tr>
<tr>
<td></td>
<td>r_boardcnf_get_brd_clk()</td>
</tr>
<tr>
<td></td>
<td>r_boardcnf_get_ddr_mbps()</td>
</tr>
</tbody>
</table>

Table 1-1  Source Files that are Not to be Modified by the User

Table 1-2  Source Files that Require Modification by Users to Suit their Hardware
2. Customizing boot_init_dram_config.c/ boot_init_dram_config.h

**Initialization Program of the User**

- **InitDram(void)**
  - Source code that is not to be modified by the user
  - `boot_init_dram.c`

- **GET Device Type**
  - The board type and configuration of the LPDDR4 SDRAM on the board are returned.
  - `r_boardcnf_get_brd_type()`

- **GET Board Config**
  - The frequency of the source clock (EXTAL) for the system PLL is returned.
  - `r_boardcnf_get_brd_clk()`

- **GET Clock Config**
  - Access rate of the on-board DRAM is returned.
  - `r_boardcnf_get_ddr_mbps()`

- **GET DRAM Speed**

**DBSC/PHY Initialize**

- **Init. Fail?**
  - Yes
    - End Return (FAIL)
  - No
    - End Return (PASS)

**Return to Further User System Initialization**

Note (*)
The 'InitDram()' function does not detect whether the DRAM specified in the configuration matches that actually mounted on the board. Accordingly, "boot_init_dram_config.c" must be correctly customized to enable the detection.

Remark
The sample source code we provide includes code that is specific to the LPDDR4X SDRAM configuration for the Renesas evaluation boards. The minimum requirements for using the DRAM initialization software are a single type of board and full information on the configuration of the LPDDR4X SDRAM mounted on the board.

Figure 2-1 Customizing boot_init_dram_config.c
2.1 Customizing boot_init_dram_config.c/ boot_init_dram_config.h

2.1.1 Setting of board configuration number

The source code can be customized to suit the hardware by modifying a structure and define directives. This section describes how to customize the source code to suit your hardware environment.

boot_init_dram_config.h

```c
/*! @details Average Interval of Periodic-WriteDQ Training[us]
 * #define REWT_TRAINING_INTERVAL 20000 /* Periodic-WriteDQ Training Interval [us] */
 */

/* NUMBER OF BOARD CONFIGURATION */
/* PLEASE DEFINE */

/*******************************************************************************
*    NUMBER OF BOARD CONFIGRATION
*    PLEASE DEFINE
******************************************************************************/

#define BOARDNUM 3U

/* Please set board number or board judge function */
#define BOARD_JUDGE_AUTO

#ifdef BOARD_JUDGE_AUTO

uint32_t r_boardcnf_get_brd_type(void) {
    return r_board_judge();
}
#else /* BOARD_JUDGE_AUTO */

uint32_t r_boardcnf_get_brd_type(void) {
    return (0);
}
#endif /* BOARD_JUDGE_AUTO */
```

(1) "BOARDNUM" is used in the boardcnfs[] structure, which holds a description of the board configuration. In the sample source code we provide, the Renesas evaluation boards are defined as values from 0 to 2.
0 and 1 are for Renesas evaluation board configuration, 2 is for general board configuration which is without signal swap.

(2) If you do not intend to automatically identify the board, comment out this define directive.
2.1.2 Definition of R-Car S4 16bit LPDDR4X SDRAM I/F CH number

R-Car S4 has two 16bit data width LPDDR4X SDRAM I/F, and 32bit width data bus is constructed by combining these two I/F. These 16bit LPDDR4X SDRAM I/F is independent each other, the LPDDR4X SDRAM initialization SW treats them as CH0 and CH1. In this document, these CH0 and CH1 are used for explanation.

On the other hands, the pin name of R-Car S4 is expressed as 32bit width data bus style, Relationship between R-Car S4 pin name and CH signal name in the initialization SW is as following table.

<table>
<thead>
<tr>
<th>CH</th>
<th>Expression of Initialization SW</th>
<th>R-Car S4 LPDDR4X SDRAM IF pin name (Only related to initialization SW )</th>
</tr>
</thead>
<tbody>
<tr>
<td>CH0</td>
<td>CA[5:0], CS0, CS1</td>
<td>M0CA[5:0]B, M0CS0B#, M0CS1B#</td>
</tr>
<tr>
<td></td>
<td>DQS[1], DQS[1]#</td>
<td>M0DQS[2], M0DQS[2]# *Note</td>
</tr>
<tr>
<td></td>
<td>DQ[7:0], DM[0]</td>
<td>M0DQ[31:24], M0DM[3] *Note</td>
</tr>
<tr>
<td></td>
<td>DQS[0], DQS[0]#</td>
<td>M0DQS[3], M0DQS[3]# *Note</td>
</tr>
<tr>
<td>CH1</td>
<td>CA[5:0], CS0, CS1</td>
<td>M0CA[5:0]A, M0CS0A#, M0CS1A#</td>
</tr>
<tr>
<td></td>
<td>DQ[15:8], DM[1]</td>
<td>M0DQ[15:8], M0DM[1]</td>
</tr>
<tr>
<td></td>
<td>DQS[1], DQS[1]#</td>
<td>M0DQS[1], M0DQS[1]#</td>
</tr>
<tr>
<td></td>
<td>DQ[7:0], DM[0]</td>
<td>M0DQ[7:0], M0DM[0]</td>
</tr>
<tr>
<td></td>
<td>DQS[0], DQS[0]#</td>
<td>M0DQS[0], M0DQS[0]#</td>
</tr>
</tbody>
</table>

Note:
It indicates that R-CarS4 LPDDR SDRAM pins seen from the CH0 have been byte-lane swapped in SoC internally.
2.1.3 Customization of boardcnf[] structure (1)

This section describes the procedure for customizing the boardcnfs[] structure by taking boardcnfs[0] as an example.

```c
static const st_boardcnf_t boardcnfs[BOARDNUM] = {
  // @details  * boardcnf[0] RENESAS S4 Spider (32Gbit 2rank)
  { 0x03,          /* phyvalid */ (*1)  
     0x0000,        /* vref_r */ (*2)  
     0x0000,        /* vref_w */ (*3)  
     0x0000,        /* vref_ca */ (*4)  
     { /* ch[0] */  
         { /* ddr_density[] */ { 0x04, 0x04 },  
         /* ca_swap */ 0x243510U,  
         /* dq_swap */ 0x10,  
         /* dq_swap[] */ { 0x21706345, 0x23510746 },  
         /* dm_swap[] */ { 0x08, 0x08 }  
       },  
       { /* ch[1] */  
         { /* ddr_density[] */ { 0x04, 0x04 },  
         /* ca_swap */ 0x345210U,  
         /* dq_swap */ 0x10,  
         /* dq_swap[] */ { 0x30124675, 0x53126047 },  
         /* dm_swap[] */ { 0x08, 0x08 }  
       }  
    },  
  
  (*1) phyvalid
  This defines the number of 16-bit external bus channels that are present.
  0x03: 16bit x 2ch, ch0 and ch1 (32bit)

  Configuration other than above are prohibited. The detailed configuration of each channel is defined in a corresponding sub-structure.

  (*2) vref_r
  This defines the start value and the stop value during VREF training of data read (SoC side VREF). When 0x0000 is defined, the start value and stop value is set SW default value. When the start value and the stop value are set same fixed value, the VREF value will become this fixed value. Detail about SoC side VREF is not opened, this configuration should be set 0x0000 unless otherwise specified from Renesas.

  (*3) vref_w
  This defines the start value and the stop value during VREF training of data write (DRAM side MR14). When 0x0000 is defined, the start value and stop value is set SW default value. When the start value and the stop value are set same fixed value, the VREF value will become this fixed value.

    [7:0]: VREF start value for training
    [15:8]: VREF stop value for training

  The stop value bigger than or same as the start value (if same VREF will become fixed value). If the range of VREF start and stop is narrow, VREF setting may not be suitable. This configuration should be set 0x0000 unless otherwise specified from Renesas.
```
(*4) vref_ca
This defines the start value and the stop value during VREF training of address and command (DRAM side MR12). When 0x0000 is defined, the start value and stop value is set SW default value. When the start value and the stop value are set same fixed value, the VREF value will become this fixed value.

[7:0]: VREF start value for training
[15:8]: VREF stop value for training

The stop value bigger than or same as the start value (if same VREF will become fixed value). If the range of VREF start and stop is narrow, VREF setting may not be suitable. This configuration should be set 0x0000 unless otherwise specified from Renesas.
2.1.4 Customization of boardcnf[] structure (2)

This section describes the procedure for customizing the boardcnfs[] structure by taking boardcnfs[0] as an example.

```c
static const st_boardcnf_t boardcnfs[BOARDNUM] = {
    0x03,  /* phyvalid */
    0x0000, /* vref_r */
    0x0000, /* vref_w */
    0x0000, /* vref_ca */
    
    /* ch[0] */ {
        /* M0CAxB/M0DQ[23:16],M0DQ[31:24] */
        /* ddr_density[] */ { 0x04, 0x04 },
        /* ca_swap */ 0x243510U,
        /* dqs_swap */ 0x10,
        /* dq_swap[] */ { 0x21706345, 0x23510746 },
        /* dm_swap[] */ { 0x08, 0x08 }
    },
    
    /* ch[1] */ {
        /* M0CAxA/M0DQ[15: 0] */
        /* ddr_density[] */ { 0x04, 0x04 },
        /* ca_swap */ 0x345210U,
        /* dqs_swap */ 0x10,
        /* dq_swap[] */ { 0x30124675, 0x53126047 },
        /* dm_swap[] */ { 0x08, 0x08 }
    }
};
```

(*1) SoC DRAM IF sub-structure for each CH
R-Car S4 does not recognize which of LPDDR4X ch(A) and ch(B) is connected to R-Car S4 CH0 and CH1. Configuration of sub-structure should be set in LPDDR4X SDRAM 16bit channel unit.

(*2) ddr_density[]

`ddr_density[]` is used to set the DRAM capacity. Configure for each RANK

First field: Capacity of the rank-1 (CS0) DRAM die
Second field: Capacity of the rank-2 (CS1) DRAM die

Capacity of LPDDR4X SDRAM x16 each channel should be set for above field.

<table>
<thead>
<tr>
<th>Value</th>
<th>Capacity</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00</td>
<td>2Gbit</td>
</tr>
<tr>
<td>0x01</td>
<td>3Gbit</td>
</tr>
<tr>
<td>0x02</td>
<td>4Gbit</td>
</tr>
<tr>
<td>0x03</td>
<td>6Gbit</td>
</tr>
<tr>
<td>0x04</td>
<td>8Gbit</td>
</tr>
<tr>
<td>0x05</td>
<td>12Gbit</td>
</tr>
<tr>
<td>0xFF</td>
<td>No memory</td>
</tr>
</tbody>
</table>

Other: setting prohibited

In above sample source, 32Gbit (8 Gbit × 2 channel × 2 rank) LPDDR4X SDRAM is used. Its ch(A): 8Gbit × 1 channel × 2 rank is connected to SoC CH1 (x16 bit), and ch(B) 8Gbit × 1 channel × 2 rank is connected to SoC CH0 (x16 bit). Total capacity of CH0 and CH1 is 4Gbyte.
2.1.5 Customization of boardcnf[] structure (3)

This section describes the procedure for customizing the boardcnfs[] structure by taking boardcnfs[0] as an example.

```c
static const st_boardcnf_t boardcnfs[BOARDNUM] = {
    /* boardcnf[0] RENESAS S4 Spider (32Gbit 2rank) */
    { 0x03, /* phyvalid */
      0x0000, /* vref_r */
      0x0000, /* vref_w */
      0x0000, /* vref_ca */
      /* ch[0] */
        { /* M0CAxB/M0DQ[23:16],M0DQ[31:24] */
          /* ddr_density[] */
            { 0x04, 0x04 },
          /* ca_swap */
            0x243510U, /*(1) */
          /* dq_swap */
            0x10,
        /* dq_swap[] */
            { 0x21706345, 0x23510746 },
        /* dm_swap[] */
            { 0x08, 0x08 },
      },
    /* ch[1] */
        { /* M0CAxA/M0DQ[15: 0] */
          /* ddr_density[] */
            { 0x04, 0x04 },
          /* ca_swap */
            0x345210U, /*(1) */
          /* dq_swap */
            0x10,
        /* dq_swap[] */
            { 0x30124675, 0x53126047 },
        /* dm_swap[] */
            { 0x08, 0x08 }
      }
    },
};
```

*1) ca_swap

ca_swap is used to specify swapping of the CA pins.

R-Car S4 PHY supports swapping of the CA signal pins in the same group. This field must be customized to suit the connection of the CA signals.

- Bit [3:0]: CA signal number of the LPDDR4 SDRAM connected to the CA0 pin of the SoC
- Bit [7:4]: CA signal number of the LPDDR4 SDRAM connected to the CA1 pin of the SoC
- Bit [11:8]: CA signal number of the LPDDR4 SDRAM connected to the CA2 pin of the SoC
- Bit [15:12]: CA signal number of the LPDDR4 SDRAM connected to the CA3 pin of the SoC
- Bit [19:16]: CA signal number of the LPDDR4 SDRAM connected to the CA4 pin of the SoC
- Bit [23:20]: CA signal number of the LPDDR4 SDRAM connected to the CA5 pin of the SoC
- Bit [31:24]: Set this field to be ‘0’.

In case of following figure, the ca_swap field is 0x00_5_0_3_4_1_2 = 0x00503412

![Figure 2-2 example of ca_swap](image-url)
2.1.6 Customization of boardcnf[] structure (4)

This section describes the procedure for customizing the boardcnfs[] structure by taking boardcnfs[0] as an example.

```c
static const st_boardcnf_t boardcnfs[BOARDNUM] = {
  //! @details  * boardcnf[0] RENESAS S4 Spider (32Gbit 2rank)
  {0x03,          /* phyvalid */
   0x0000,        /* vref_r */
   0x0000,        /* vref_w */
   0x0000,        /* vref_ca */
   {/* ch[0] */  /* M0CAxB/M0DQ[23:16],M0DQ[31:24] */
      /* ddr_density[] */ { 0x04, 0x04 },
      /* ca_swap */  0x243510U,
      /* dqs_swap */  0x10,     (*1)
      /* dq_swap[] */  { 0x21706345, 0x23510746 },
      /* dm_swap[] */  { 0x08, 0x08 }
   },
   {/* ch[1] */  /* M0CAxA/M0DQ[15: 0] */
      /* ddr_density[] */ { 0x04, 0x04 },
      /* ca_swap */  0x345210U,
      /* dqs_swap */  0x10,     (*1)
      /* dq_swap[] */  { 0x30124675, 0x53126047 },
      /* dm_swap[] */  { 0x08, 0x08 }
   }
  }
```

(*1) dqs_swap

Byte-lane swap configuration in 16bit width LPDDR4X SDRAM IF,

R-Car S4 supports byte-lane swapping in LPDDR4X SDRAM ch(A) and ch(B), This field should be customized to suit the connection of the DQS signals.

Bit [3:0]: LPDDR4X SDRAM byte lane number connected to byte lane 0 (DQS0) of the SoC
Bit [7:4]: LPDDR4X SDRAM byte lane number connected to byte lane 1 (DQS1) of the SoC
Bit [15:8]: Set this field to be ‘0’

Definition byte-lane number for LPDDR4X SDRAM ch(A) and ch(B) is as follow.

- DQ7-DQ0, DM10, DQS0 / DQS0#: byte lane number = 0
- DQ15-DQ8, DM11, DQS1 / DQS1#: byte lane number = 1

Remark: R-Car S4 LPDDR4X SDRAM pins seen from CH0 have been swapped in SoC internally. For detail, please refer to the section 2.1.2
<table>
<thead>
<tr>
<th></th>
<th>R-Car S4</th>
<th>LPDDR4X SDRAM</th>
<th></th>
<th>R-Car S4</th>
<th>LPDDR4X SDRAM</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>dqs_swap =0x10</strong></td>
<td></td>
<td></td>
<td><strong>dqs_swap =0x01</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DM0</td>
<td>DMI0</td>
<td>DM0</td>
<td>DQ50/DQ50#</td>
<td>DQ50/DQ50#</td>
<td>DQ50/DQ50#</td>
</tr>
<tr>
<td>DM1</td>
<td>DMI1</td>
<td>DM1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DQ51/DQ51#</td>
<td></td>
<td>DQ51/DQ51#</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Above signal name of R-Car S4 side is name for initialization SW, not for pin name.

**Figure 2-3 dqs_swap setting**
2.1.7 Customization of boardcnf[] structure (5)

This section describes the procedure for customizing the boardcnfs[] structure by taking boardcnfs[0] as an example.

```c
static const st_boardcnf_t boardcnfs[BOARDNUM] = {
  //! @details * boardcnf[0] RENESAS S4 Spider (32Gbit 2rank)
  { 0x03,          /* phyvalid */
         0x0000,        /* vref_r */
         0x0000,        /* vref_w */
         0x0000,        /* vref_ca */
         { /* ch[0] */
            { /* M0CAXB/M0DQ[23:16],M0DQ[31:24] */
              { /* ca_swap */        0x243510U,
                { /* dqs_swap */        0x10,
                  { /* dq_swap[] */        { 0x21706345, 0x23510746 },
                    { /* dm_swap[] */        { 0x08, 0x08 },
                      { /* ch[1] */
                        { /* M0CAxA/M0DQ[15:0] */
                          { /* ca_swap */        0x345210U,
                            { /* dqs_swap */        0x10,
                              { /* dq_swap[] */        { 0x30124675, 0x53126047 },
                                { /* dm_swap[] */        { 0x08, 0x08 }
                              },
                        },
                      },
                    },
                  },
                },
              },
            },
          },
  },
```

(*1) dq_swap

(*2) dm_swap

dq_swap and dm_swap are used to configure swapping of the DQ and DM signals within the byte-lane.

R-Car S4 supports swapping of the DQ and DM signals within the byte lanes. This field is used to set the DQ pins within the DRAM byte lane to be connected to the DQ pins within the R-Car S4 byte-lane. The order of the byte lanes is from byte lane 0 to byte lane 1.

dq_swap

Bit [3:0]: DQ pin within the LPDDR4X SDRAM byte-lane to be connected to the DQ0 pin within the SoC byte lane.

Bit [7:4]: DQ pin within the LPDDR4X SDRAM byte-lane to be connected to the DQ1 pin within the SoC byte lane.

Bit [11:8]: DQ pin within the LPDDR4X SDRAM byte-lane to be connected to the DQ2 pin within the SoC byte lane.

Bit [15:12]: DQ pin within the LPDDR4X SDRAM byte-lane to be connected to the DQ3 pin within the SoC byte lane.

Bit [19:16]: DQ pin within the LPDDR4X SDRAM byte-lane to be connected to the DQ4 pin within the SoC byte lane.

Bit [23:20]: DQ pin within the LPDDR4X SDRAM byte-lane to be connected to the DQ5 pin within the SoC byte lane.

Bit [27:24]: DQ pin within the LPDDR4X SDRAM byte-lane to be connected to the DQ6 pin within the SoC byte lane.

Bit [31:28]: DQ pin within the LPDDR4X SDRAM byte-lane to be connected to the DQ7 pin within the SoC byte lane.

dm_swap

Bits [3:0]: DM pin within the LPDDR4X SDRAM byte-lane to be connected to the DM pin within the SoC byte lane.

Bits [7:4]: Set 0 in this field

Note:
The DQ [15:8] of LPDDR4X SDRAM are replaced as DQ[7:0]
The DMI of LPDDR4X SDRAM is replaced as DQ8 pin
When following figure, the configuration setting is as follow.

dq_swap field = 0x8_4_0_6_1_5_3_2 = 0x84061532

dm_swap field = 0x0_7 = 0x07

Remark: R-Car S4 LPDDR4X SDRAM pins seen from CH0 have been swapped in SoC internally. For detail, please refer to the section 2.1.2
2.2 LPDDR4X SDRAM refresh rate adjusting

Generally, in order to maintain DRAM contents under high temperature, higher rate refresh is needed. According to JEDEC specification, two kind of refresh rate are defined for high temperature, but its border of temperature are vendor specific. Renesas recommends using highest refresh rate statically under highest temperature that allowed by the system. Of course, used LPDDR4X SDRAM specification is satisfied with this highest temperature.

Table 2-2 Part of JEDEC LPDDR4 SDRAM MR4 Specification

<table>
<thead>
<tr>
<th>Refresh rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 x refresh (default)</td>
</tr>
<tr>
<td>0.5 x refresh</td>
</tr>
<tr>
<td>0.25 x refresh, no de-rating</td>
</tr>
<tr>
<td>0.25 x refresh, with de-rating</td>
</tr>
</tbody>
</table>

Refresh rate setting of DRAM initialization is used for refresh interval during executing initialization by this SW, this refresh interval setting during DRAM initialization may be set again by program after initialization, but when maintain DRAM contents during DRAM initialization (DRAM back-up etc), these configuration should be set suitable value.

There are two parameters for setting of refresh interval. One is average refresh interval time setting (DBRFCNF1:REFINT), and the other is refresh interval adjustment (DBRFCNF2:REFINTS, REFINT x1 or REFINT x 1/2)

```c
#define JS2_DERATE 0
#define DBSC_REFINT 1920  /* Average periodic refresh interval/Average Refresh Interval [ns] */
#define DBSC_REFINTS 0  /* 0: Average interval is REFINT. / 1: Average interval is 1/2 REFINT. */
#define REWT_TRAINING_INTERVAL 20000  /* Periodic-WriteDQ Training Interval [us] */
```

Modifying code for DBRFCNF2 in boot_init_dram_regdef.h

REFINT and REFINTS initial value (green hatched)
2.3 LPDDR4X SDRAM temperature derating configuration

This configuration targets for adding extra timing margin at 85 deg.C to 105 deg.C which is based on JEDEC Standard 209-4B Table 131 (Temperature Derating AC Timing)

```
#define JS2_DERATE 0
#define DBSC_REFINT 1920 /* Average periodic refresh interval/Average Refresh Interval [ns] */
#define DBSC_REFINTS 0 /* 0: Average interval is REFINT. / 1: Average interval is 1/2 REFINT. */
#define REWT_TRAINING_INTERVAL 20000 /* Periodic-WriteDQ Training Interval [us] */
```

Modifying code for temperature derating configuration in boot_init_dram_regdef.c (green hatched)

When JS_DERATE is set to 0, temperature derating is not valid. When JS_DERATE is not set to 0, temperature derating is valid. Whether temperature derating is necessary or not in use-case temperature, please ask DRAM vendor. When temperature derating is valid, access performance may deteriorate in order to guarantee timing margin in high temperature.
2.4 Periodic Write DQ training interval configuration

Due to temperature changes and/or power supply voltage changes during operation, the timing relationship between data (DQ) and strobe signal (DQS) may shift and become non-suitable. The periodic write DQ training is training to readjust timing during system operation, and it is executed periodically during operation like auto-refresh.

The configuration for periodic write DQ training interval sets the training interval in micro-sec units.

```c
#define JS2_DERATE 0
#define DBSC_REFINT 1920  /* Average periodic refresh interval/Average Refresh Interval [ns] */
#define DBSC_REFINTS 0  /* 0: Average interval is REFINT. / 1: Average interval is 1/2 REFINT. */
#define REWT_TRAINING_INTERVAL 20000  /* Periodic-WriteDQ Training Interval [us] */
```

During executing of periodic write DQ training, DRAM access cannot perform, therefore, if the training interval is too short, degradation of DRAM bus performance will become conspicuous. On the other hands, if it is too long, risk of degradation of DRAM access margin will increase, therefore, its balance is important.

Periodic write DQ training interval is related to the QoS setting of DRAM bus, and the setting is often reconfigured by the upper layer S/W. This configuration should be matched with that upper layer S/W. Regarding the setting and its way of the upper S/W side, refer to the manual of the upper layer S/W.
2.5 Customizing void boardcnf_get_brd_clk() function

```c
/**
 * EXTAL CLOCK DEFINITION
 * PLEASE DEFINE HOW TO JUDGE BORAD CLK
 */
/**
 * RENESAS SPIDER BOARD EXAMPLE
 * @par TraceID Cover_SW_UD:Cover_HW_DD
 * S4-DDR-SW-UD-01-04-01-02:S4-DDR-HW-DD-01-04-01
 * @param[in] brd Argument for dummy read
 * @param[in] clk The pointer which indicate the clock frequency
 * @param[in] div The pointer which indicate the clock frequency division
 * @details judged by md14/md13
 * 16.00MHz CLK,DIV= 48,3  (md14,md13==0,0)
 * 20.00MHz CLK,DIV= 60,3  (md14,md13==0,1)
 * 40.00MHz CLK,DIV=120,3  (md14,md13==1,1)
 */
void r_boardcnf_get_brd_clk(uint32_t brd, uint32_t *clk, uint32_t *divi) {
    uint32_t md;
    md = (mmio_read_32(RST_MODEMR0) >> 13) & 0x3U;
    switch(md) {
        case 0x0 : *clk = 48; *divi = 3; break;              /* 48 / 3 = 16.00MHz */
        case 0x1 : *clk = 60; *divi = 3; break;              /* 60 / 3 = 20.00MHz */
        case 0x2 : NOTICE("MD14/MD13 is invalid!!\n");panic; /* Not supported */
        case 0x3 : *clk =120; *divi = 3; break;              /* 120 / 3 = 40.00MHz */
    }
    (void)brd;
}
```

This function returns the frequency of the EXTAL clock (system clock) connected to the SoC.

The EXTAL frequency is defined by CLK (*clk) / DIV (*div), both integers. The sample source code we provide uses the states of the MD pins to determine the frequency of the EXTAL clock. When the EXTAL clock connected to the board is different from that for which the function in the sample code is written, modify this function.

Note:
When different frequency EXTAL clock from described in the R-Car S4 hardware manual, please ask Renesas whether it is no problem or not,
2.6 Customizing void boardcnf_get_ddr_mbps() function

```c
void r_boardcnf_get_ddr_mbps(uint32_t brd, uint32_t *mbps, uint32_t *divi) {
    uint32_t md;
    uint32_t sscg;

    md = (mmio_read_32(RST_MODEMR0) >> 17) & 0x01U;
    sscg = (mmio_read_32(RST_MODEMR1) >> 4) & 0x03U;

    switch(sscg) {
        case 0x0:
            switch(md) {
                case 0x0: *mbps = 3200; *divi = 1; break;
                case 0x1: *mbps = 2133; *divi = 1; break;
            }
            break;
        case 0x1:
            switch(md) {
                case 0x0: *mbps = 3120; *divi = 1; break;
                case 0x1: *mbps = 2133; *divi = 1; break;
            }
            break;
        case 0x2:
            switch(md) {
                case 0x0: *mbps = 3040; *divi = 1; break;
                case 0x1: *mbps = 2133; *divi = 1; break;
            }
            break;
        case 0x3:
            switch(md) {
                case 0x0: *mbps = 3000; *divi = 1; break;
                case 0x1: *mbps = 2133; *divi = 1; break;
            }
            break;
    }
    (void)brd;
}
```

This function returns the access rate of the target LPDDR4 SDRAM, defined by MBPS(*mbps) / DIV(*div) in units of Mbps.

Setting of PLL3 (LPDDR4X SDRAM access speed) VCO frequency has limitation in relationship with PLL1 VCO frequency. When SSCG function of PLL1 is enabled, the target is minimum frequency of PLL1, for detail, please refer to the R-Car S4 hardware manual.

DRAM access speed in the sample source code is defined by PLL1 minimum frequency and multiple rate of PLL3 VCO.
Acceptable LPDDR4X SDRAM access speed (\*mbps) is as follow.

- **3200Mbps series**  
  from 2880Mbps to 3200Mbps [EXTAL = 40/20MHz case]  
  from 2944Mbps to 3200Mbps [EXTAL = 16MHz case]

- **2133Mbps series**  
  from 2000Mbps to 2133Mbps [EXTAL = 40/20MHz case]  
  from 2048Mbps to 2133Mbps [EXTAL = 16MHz case]

The setting out of above support range is prohibit.
## Revision History

<table>
<thead>
<tr>
<th>Revision</th>
<th>Date</th>
<th>History</th>
<th>Page</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>10 Dec. 2021</td>
<td>-</td>
<td>P.7</td>
<td>2.1.3 Description 'This configuration should be set 0x0000 unless otherwise specified from Renesas.' were added to vref_w and vref_ca explanation.</td>
</tr>
</tbody>
</table>
| 1.01     | 1 Feb, 2022  | P.6     | 2.1.2| • Table 2-1, expression for DQ signals were changed to byte-lane unit  
• Table 2-1, Following note was added.  
"It indicates that R-CarS4 LPDDR SDRAM pins seen from the CH0 have been byte-lane swapped in SoC internally." |
| 1.02     | 4 Mar, 2022  | P.11    | 2.1.2| • Following remark was added to explanation for dqs_swap  
"R-CarS4 LPDDR SDRAM pins seen from the CH0 have been byte-lane swapped in SoC internally. For detail, please refer to 2.1.2 section." |
| P.13     |              | P.13    | 2.1.7| • Explanation above figure 2-4 was modified because it was not matched with the figure.  
• LPDDR4 in figure 2-4 was changed to LPDDR4X  
• Following remark was added to explanation for dq_swap and dm_swap  
"R-CarS4 LPDDR SDRAM pins seen from the CH0 have been byte-lane swapped in SoC internally. For detail, please refer to 2.1.2 section." |
| P.17, P.18 |            | P.18    | 2.3.2| • Sample source for explanation was updated to the latest one (rev.1.01rc2)  
• Explanation was modified, and new explanation was added for 16MHz EXTAL case.  
(Before modification)  
DRAM access speed are integer value which is divisible by 40Mbps  
(After modification)  
DRAM access speed are integer value which is divisible by 80Mbps\textsuperscript{Note}  
• The selectable frequency range was modified, and frequency range in case of 16MHz EXTAL were added. |
<table>
<thead>
<tr>
<th>Revision</th>
<th>Date</th>
<th>History</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.03</td>
<td>15 June, 2022</td>
<td>P.1</td>
<td>Target source code revision was changed from rev.0.02 to rev.0.20</td>
</tr>
<tr>
<td></td>
<td></td>
<td>P.4</td>
<td>Figure 2-1 Expression of return from initialization were modified.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>P.17</td>
<td>The source code was updated to rev.0.20.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>P.18</td>
<td>Following description and related note were deleted.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>‘In particular, the DRAM access speed are integer value which is divisible by 80Mbps and not to exceed the minimum frequency of PLL1 VCO.’</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>The maximum access speed of 2133Mbps setting was modified from 2080Mbps to 2120Mbps.</td>
</tr>
<tr>
<td>1.04</td>
<td>1 October, 2022</td>
<td>P.1</td>
<td>Target source code revision was changed from rev.0.02 to rev.0.40</td>
</tr>
<tr>
<td></td>
<td></td>
<td>P.5, P.18</td>
<td>Sample source code was changed to rev.0.40.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>P.9</td>
<td>12Gbit and 16Gbit setting were added for DRAM density configuration</td>
</tr>
<tr>
<td></td>
<td></td>
<td>P.16</td>
<td>Explanation about the periodic write DQ training interval setting was added</td>
</tr>
<tr>
<td></td>
<td></td>
<td>P.17, P.18</td>
<td>Chapter number were changed. 2.3.1 to 2.5, 2.3.2 to 2.6.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>P.19</td>
<td>Support range for DRAM access speed setting was updated.</td>
</tr>
<tr>
<td>Revision</td>
<td>Date</td>
<td>History</td>
<td>Page</td>
</tr>
<tr>
<td>----------</td>
<td>------------</td>
<td>---------</td>
<td>------</td>
</tr>
<tr>
<td>1.05</td>
<td>7.January, 2023</td>
<td>P.1</td>
<td>Target source code revision was changed from rev.0.40 to rev.0.50</td>
</tr>
<tr>
<td></td>
<td></td>
<td>P.2</td>
<td>Change the title of the table of contents Chapter 2 and Chapter 2.1.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>“Customizing boot_init_dram_config.c”→ “Customizing boot_init_dram_config.c/boot_init_dram_config.h”</td>
</tr>
<tr>
<td></td>
<td></td>
<td>P.3</td>
<td>Change the description of “Table 1 2  Source Files that Require Modification by Users to Suit their Hardware”.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>boardcnf_get_brd_type()→r_boardcnf_get_brd_type()</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>boardcnf_get_ddr_mbps()→r_boardcnf_get_ddr_mbps()</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>boardcnf_get_brd_clk()→r_boardcnf_get_brd_clk()</td>
</tr>
<tr>
<td></td>
<td></td>
<td>P.4</td>
<td>Change the function name in “Figure 2 1 Customizing boot_init_dram_config.c”</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>boardcnf_get_brd_type()→r_boardcnf_get_brd_type()</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>boardcnf_get_ddr_mbps()→r_boardcnf_get_ddr_mbps()</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>boardcnf_get_brd_clk()→r_boardcnf_get_brd_clk()</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Change the title of Chapter 2 and Chapter 2.1.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>“Customizing boot_init_dram_config.c”→ “Customizing boot_init_dram_config.c/boot_init_dram_config.h”</td>
</tr>
<tr>
<td></td>
<td></td>
<td>P.5</td>
<td>Change the setting location of the board configuration number.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>P.11</td>
<td>Add explanation for bit[15:8] of dqs_swap</td>
</tr>
<tr>
<td></td>
<td></td>
<td>P.5~P.19</td>
<td>Updated each code content.</td>
</tr>
</tbody>
</table>
General Precautions in the Handling of Microprocessing Unit and Microcontroller Unit Products

The following usage notes are applicable to all Microprocessing unit and Microcontroller unit products from Renesas. For detailed usage notes on the products covered by this document, refer to the relevant sections of the document as well as any technical updates that have been issued for the products.

1. Precaution against Electrostatic Discharge (ESD)
   A strong electrical field, when exposed to a CMOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop the generation of static electricity as much as possible, and quickly dissipate it when it occurs. Environmental control must be adequate. When it is dry, a humidifier should be used. This is recommended to avoid using insulators that can easily build up static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors must be grounded. The operator must also be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions must be taken for printed circuit boards with mounted semiconductor devices.

2. Processing at power-on
   The state of the product is undefined at the time when power is supplied. The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the time when power is supplied. In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the time when power is supplied until the reset process is completed. In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the time when power is supplied until the power reaches the level at which resetting is specified.

3. Input of signal during power-off state
   Do not input signals or an I/O pull-up power supply while the device is powered off. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements. Follow the guideline for input signal during power-off state as described in your product documentation.

4. Handling of unused pins
   Handle unused pins in accordance with the directions given under handling of unused pins in the manual. The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of the LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible.

5. Clock signals
   After applying a reset, only release the reset line after the operating clock signal becomes stable. When switching the clock signal during program execution, wait until the target clock signal is stabilized. When the clock signal is generated with an external resonator or from an external oscillator during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Additionally, when switching to a clock signal produced with an external resonator or by an external oscillator while program execution is in progress, wait until the target clock signal is stable.

6. Voltage application waveform at input pin
   Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between VIL (Max.) and VIH (Min.) due to noise, for example, the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between VIL (Max.) and VIH (Min.).

7. Prohibition of access to reserved addresses
   Access to reserved addresses is prohibited. The reserved addresses are provided for possible future expansion of functions. Do not access these addresses as the correct operation of the LSI is not guaranteed.

8. Differences between products
   Before changing from one product to another, for example to a product with a different part number, confirm that the change will not lead to problems. The characteristics of a microprocessing unit or microcontroller unit products in the same group but having a different part number might differ in terms of internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.
Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of the circuits, software, or information.

2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving patents, copyrights, or other intellectual property rights of third parties, by or arising from the use of Renesas Electronics products or technical information described in this document, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.

3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.

4. You shall be responsible for determining what licenses are required from any third parties, and obtaining such licenses for the lawful import, export, manufacture, sales, utilization, distribution or other disposal of any products incorporating Renesas Electronics products, if required.

5. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, conditions are within the ranges specified by Renesas Electronics.

6. Renesas Electronics products are classified according to the following two quality grades: “Standard” and “High Quality”. The intended applications for each Renesas Electronics product depend on the product’s quality grade, as indicated below.

   - **Standard**: Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.
   - **High Quality**: Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.

   Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems; surgical implantations; etc.), or may cause serious property damage (space systems; underwater repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user’s manual or other Renesas Electronics document.

7. No semiconductor product is absolutely secure. Notwithstanding any security measures or features that may be implemented in Renesas Electronics hardware or software products, Renesas Electronics shall have absolutely no liability arising out of any vulnerability or security breach, including but not limited to, any unauthorized access to or use of a Renesas Electronics product or a system that uses a Renesas Electronics product. RENESAS ELECTRONICS DOES NOT WARRANT OR GUARANTEE THAT RENESAS ELECTRONICS PRODUCTS, OR ANY SYSTEMS CREATED USING RENESAS ELECTRONICS PRODUCTS WILL BE INVULNERABLE OR FREE FROM CORRUPTION, ATTACK, VIRUSES, INTERFERENCE, HACKING, DATA LOSS OR THEFT, OR OTHER SECURITY INTRUSION (“Vulnerability Issues”). RENESAS ELECTRONICS DISCLAIMS ANY AND ALL RESPONSIBILITY OR LIABILITY ARISING FROM OR RELATED TO ANY VULNERABILITY ISSUES. FURTHERMORE, TO THE EXTENT PERMITTED BY APPLICABLE LAW, RENESAS ELECTRONICS DISCLAIMS ANY AND ALL WARRANTIES, EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT AND ANY RELATED OR ACCOMPANYING SOFTWARE OR HARDWARE, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE.

8. When using Renesas Electronics products, refer to the latest product information (data sheets, user’s manuals, application notes, “General Notes for Handling and Using Semiconductor Devices” in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.

9. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, such as safety design for hardware and software, including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.

10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.

11. Renesas Electronics products and technologies shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.

12. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.

13. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.

14. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

(Note1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.

(Note2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

(Rev.5.0-1 October 2020)

Corporate Headquarters
TOYOSU FORESIA, 3-2-24 Toyosu, Koto-ku, Tokyo 135-0061, Japan
www.renesas.com

Trademarks
Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.

© 2023 Renesas Electronics Corporation. All rights reserved.