Renesas Synergy™ Platform

S7G2 Parity Error Workaround

Introduction

Renesas has identified an issue with the S7G2 MCU, where spurious parity error interrupts are generated for a specific SRAM address range when the ICLK frequency is above 120 MHz and parity checking is enabled.

The S7G2 has 640 KB of contiguous SRAM, and the 216-KB region that is impacted by this issue is in the middle of the SRAM range (0x20008000 to 0x2003DFFF). A workaround for this issue is to exclude the SRAM at this address range by masking off the region with a constant buffer of 216 KB so that this region is not used by the user application, thereby avoiding the spurious parity error interrupts.

Target Device

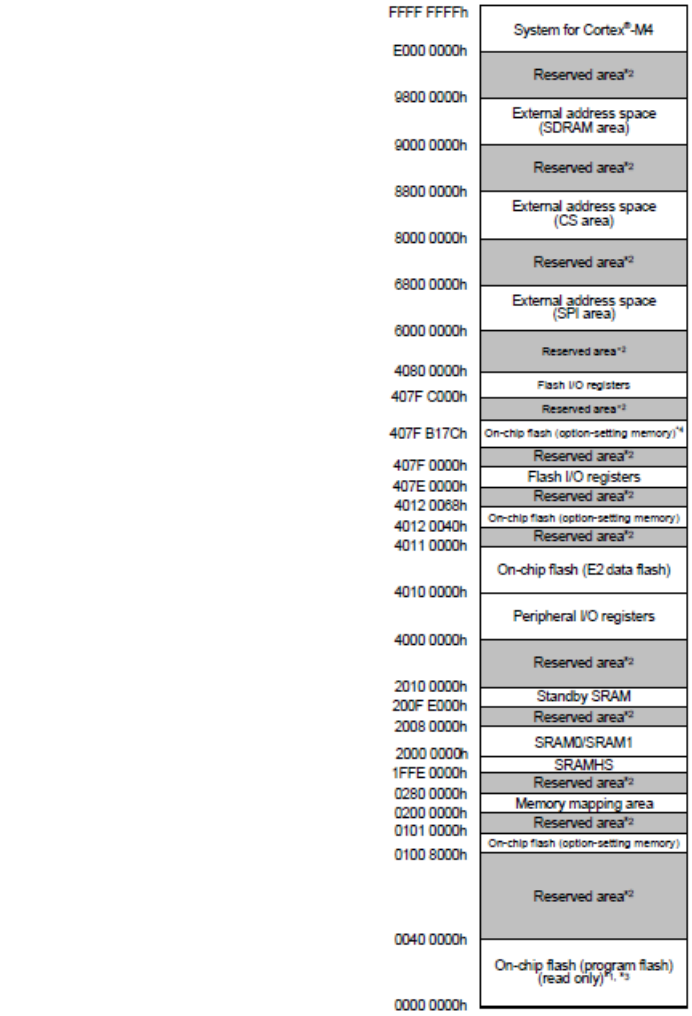
S7G2

Contents

1. Memory Organization and the SRAM address range for S7G2 2
1.1 Memory map of S7G2 MCU 2
1.2 Observed error 2
2. Linker Script Workaround for the IAR Toolchain 3
2.1 Define symbols for addresses of start and end of region in the linker script 3
2.2 Define the reserved region in the linker script 3
2.3 Declare the placeholder for the buffer in the linker script 4
2.4 Place a constant buffer in the reserved region in the application code/hal_entry.c 4
3. Linker Script Workaround for the GCC Toolchain 4
3.1 Define memory region in the linker script 4
3.2 Place a constant buffer in the reserved region in the application code/hal_entry.c 6
4. Expected output and memory allocation 7
4.1 .map file with IAR linking 7
4.2 .map file with GCC linking 7
5. Notes 7
6. References 8
Revision History 10

1. Memory Organization and the SRAM address range for S7G2

1.1 Memory map of S7G2 MCU



Note 1. The capacity of the flash differs depending on the products:

Product	Address
4-MB product	0000 0000h to 003F FFFFh
3-MB product	0000 0000h to 002F FFFFh

Note 2. Do not access reserved areas.

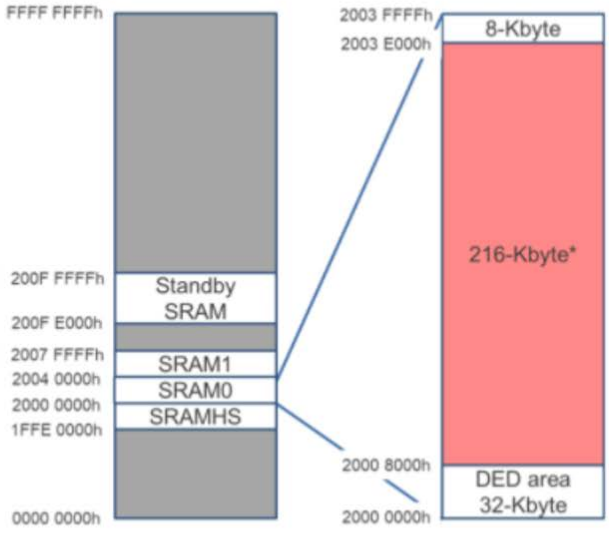
Figure 1. S7G2 Memory Map

- Contiguous SRAM region of 640 KB spans the 1FFE0000H to 20080000H address range.
- The MCU provides on-chip, high-speed SRAM modules with optional parity-bit checking or DED (double-bit error detection). The first 32 KB of SRAM0 supports DED. Parity checking is performed on other areas.

1.2 Observed error

The address region 0x20008000-0x2003DFFF is subject to spurious parity interrupts when the MCU is operated at ICLK greater than 120 MHz and when parity error interrupts or resets are enabled. Under these conditions an invalid interrupt or reset may occur.

- SRAM data can be correctly read and written up to ICLK = 240 MHz even when the parity error flag shows the wrong value.



Memory map

*Parity circuit works correctly up to ICLK 120MHz in this 216-Kbyte area

Parity circuit works correctly up to 240MHz in all the rest of the area.

Figure 2. Memory Map of Affected Area

2. Linker Script Workaround for the IAR Toolchain

As a workaround, the address range 0x20008000-0x2003DFFF can be masked off by declaring a constant buffer of 216 KB in the IAR linker script using following steps.

2.1 Define symbols for addresses of start and end of region in the linker script

Define symbols for the start and end addresses for the region to be masked as a reserved region in the linker script `S7G2.icf`

```
s7g2.icf | hal_entry.c | s7g2_iar_test.map | startup_S7G2.c | main.c | hal_entry.c | startup
14  define symbol region_FLASH_end      = 0x003FFFFF;
15  define symbol region_RAM_start      = 0x1FFE0000;
16  define symbol region_RAM_end        = 0x2007FFFF;
17  define symbol region_reserved_start = 0x20008000;
18  define symbol region_reserved_end   = 0x2003DFFF;
19  define symbol region_DF_start       = 0x40100000;
20  define symbol region_DF_end         = 0x4010FFFF;
```

Figure 3. Start and End Regions

2.2 Define the reserved region in the linker script

```
s7g2.icf
34  define region FLASH_region      = mem:[from region_FLASH_start   to region_FLASH_end];
35  define region RAM_region        = mem:[from region_RAM_start     to region_RAM_end];
36  define region DF_region         = mem:[from region_DF_start      to region_DF_end];
37  define region SDRAM_region      = mem:[from region_SDRAM_start   to region_SDRAM_end];
38  define region QSPI_region       = mem:[from region_QSPI_start    to region_QSPI_end];
39
40  define region ID_CODE_region    = mem:[from region_ID_CODES_start   to region_ID_CODES_end];
41
42  define region reserved_region     = mem:[from region_reserved_start   to region_reserved_end];
43
44  initialize manually { readwrite };
```

Figure 4. Reserved Region

2.3 Declare the placeholder for the buffer in the linker script

This will reserve the 216-KB address space as `reserved_region` starting from 0x20008000-0x2003DFFF

```
s7g2.icf
93                              rw section .data,
94                              block HW_LOCK,
95                              rw section HEAP,
96                              rw section .stack,
97                              block RAM_CODE };
98
99          place in reserved_region  {rw section BUFFER};
100
101         place in DF_region      { ro section .data_flash };
102         place in SDRAM_region   { rw section .sdram, rw section .frame };
103         place in QSPI_region    { section .qspi_flash  };
```

Figure 5. Placeholder

2.4 Place a constant buffer in the reserved region in the application code/hal_entry.c

This will place a constant buffer of 216-KB at the defined location, hence masking the region off.

```
1   /* HAL-only entry function */
2   #include "hal_data.h"

    #if defined(__ICCARM__)
    #pragma location="BUFFER"
    __root uint8_t buffer[216*1024]={0};
    #endif
```

Figure 6. Buffer in Reserved Region

3. Linker Script Workaround for the GCC Toolchain

As a workaround, the address range 0x20008000-0x2003DFFF can be masked off by declaring a const buffer of 216 KB and using the GCC linker script to place it at the desired address using following steps.

3.1 Define memory region in the linker script

Define `reserved_section` as a region in memory with start address 0x20008000 in the RAM section and define a section in this region as .buffer in the linker script file (`s7g2.ld`).

Ensure the following when placing the buffer in the GCC linker script to use all 160 KB available before address 0x20008000:

1. Make sure the buffer is placed after the stack area so as to use as much of the 160-KB memory region available before address 0x20008000 as possible.

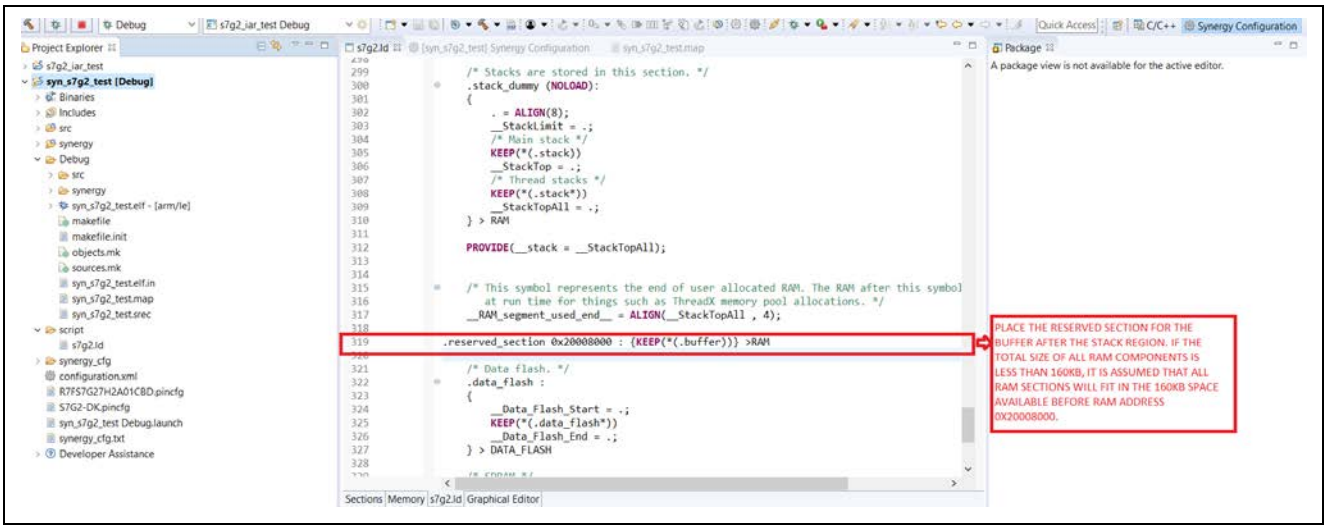


Figure 7. Define and place reserved_section

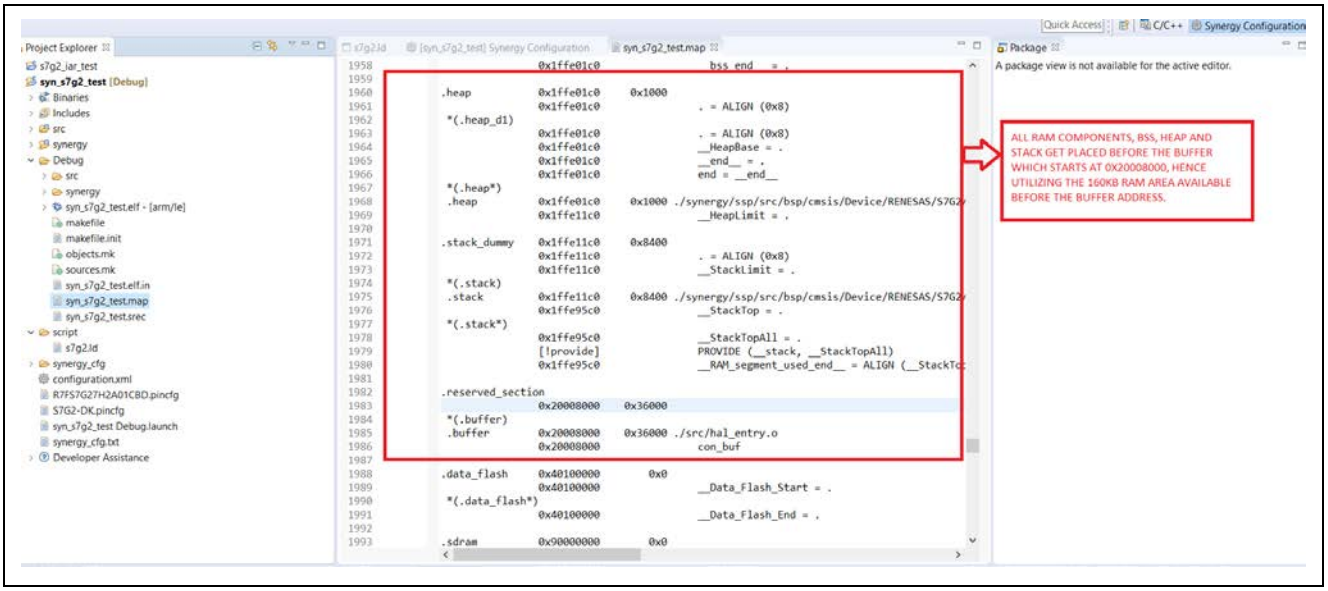


Figure 7. Map file showing the placement of the buffer and RAM sections before the buffer.

2. If any region overflows into the buffer and the linker throws an overlapping regions error, move the buffer before the region until the overflow no longer occurs. If the addition of RAM regions exceeds 160 KB, move the buffer before the region that stops the overlap error.

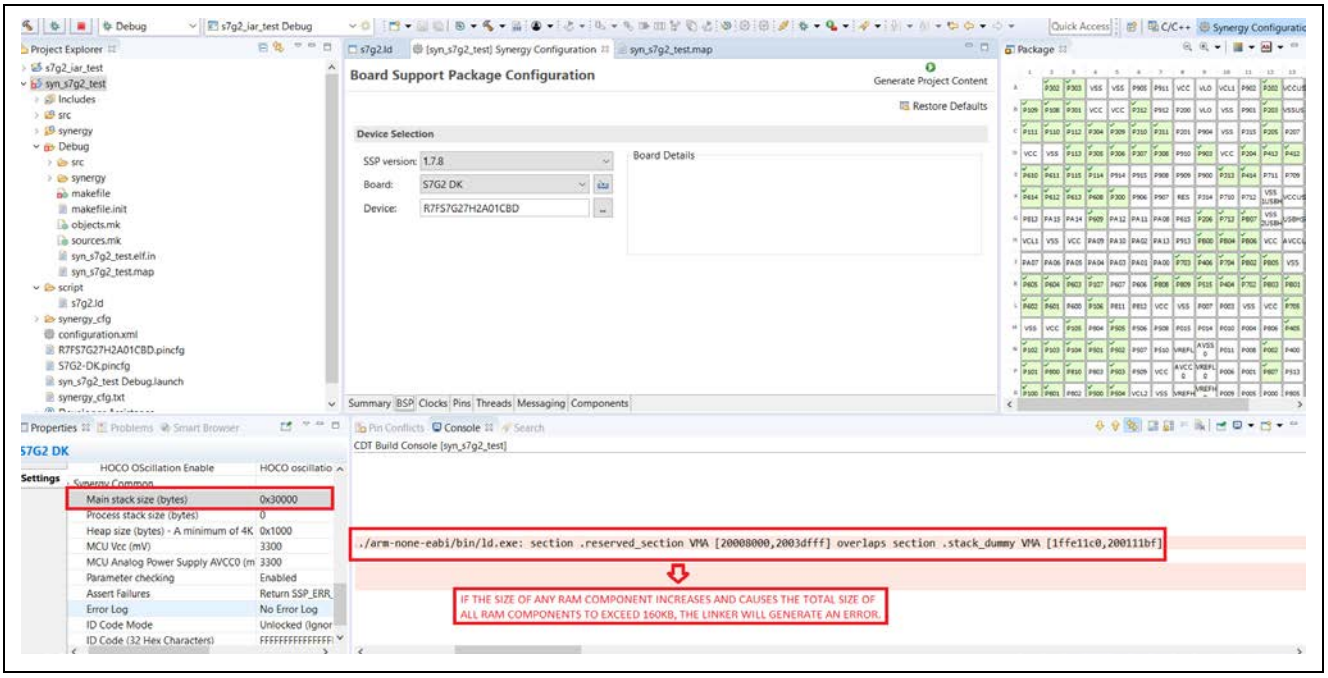


Figure 8. Stack region is increased so that it overflows into buffer and the linking error is generated

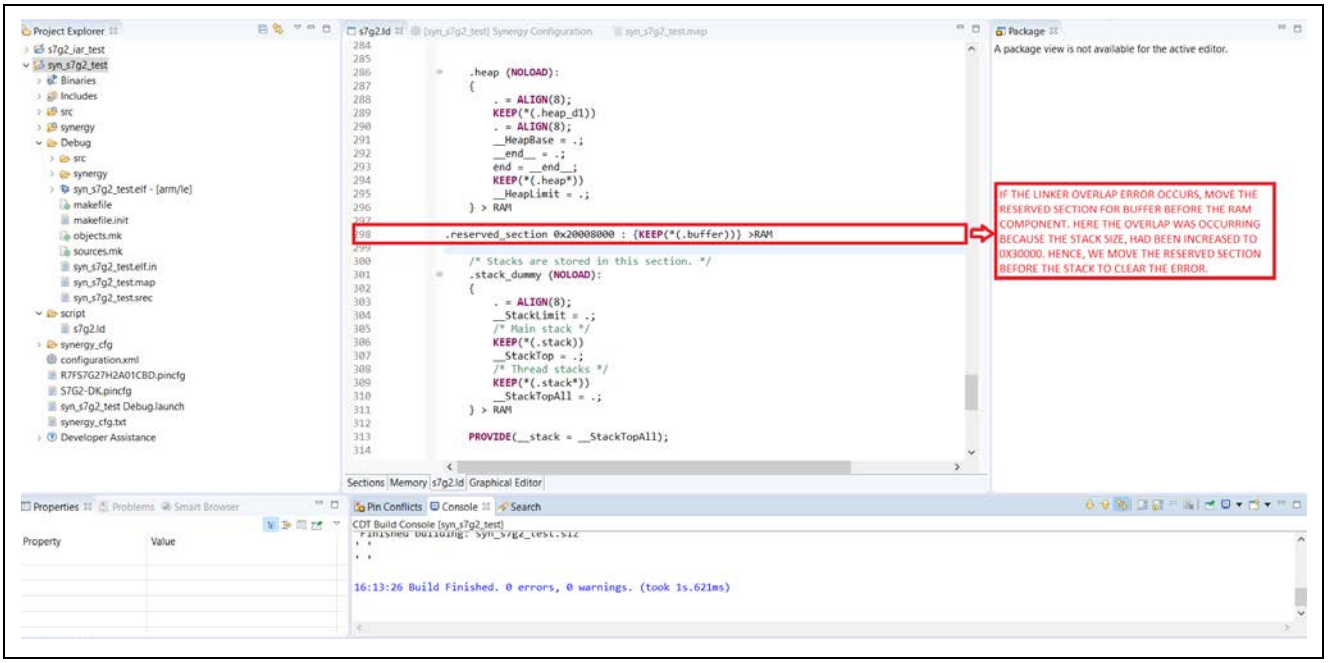


Figure 9. Buffer is moved before the stack region so that the linker error is cleared

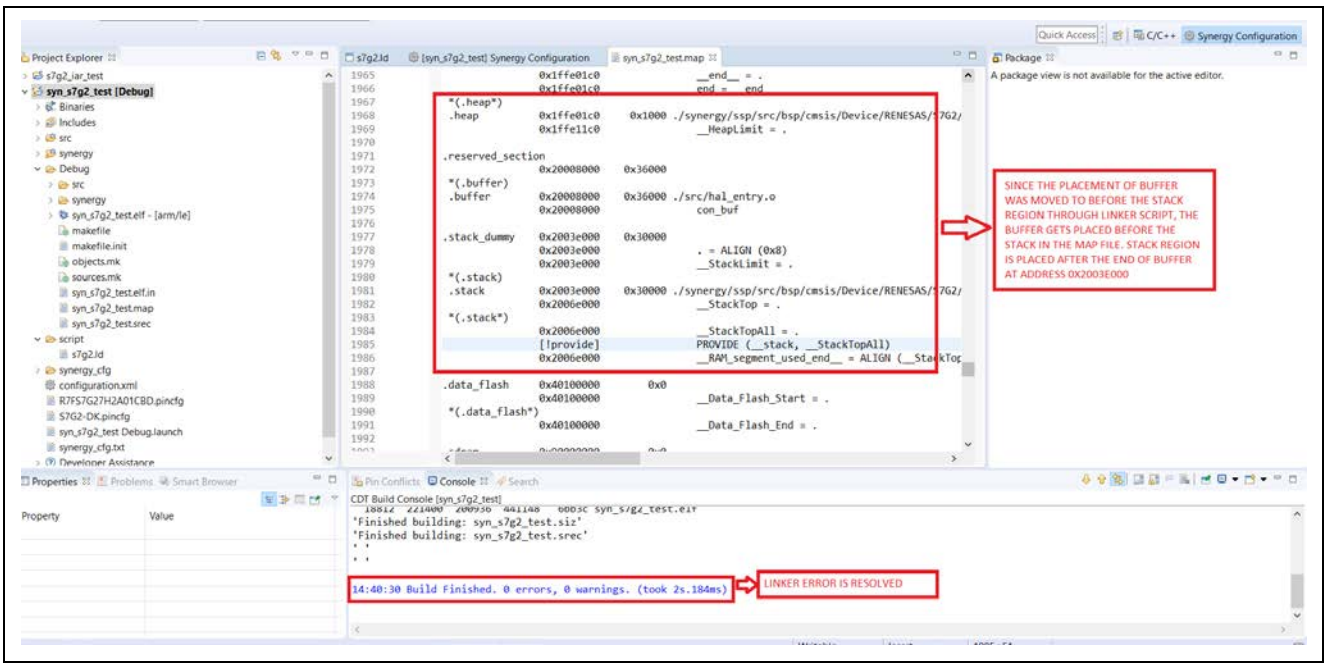


Figure 10. Buffer is placed before the stack since it was placed before the stack in the linker script

Note: The user is responsible for changing the placement of the buffer as needed. The buffer needs to be located at the right place for optimum usage of available memory before address 0x20008000.

3.2 Place a constant buffer in the reserved region in the application code/hal_entry.c

This will place a const buffer of 216 KB at the defined location, masking the region off.

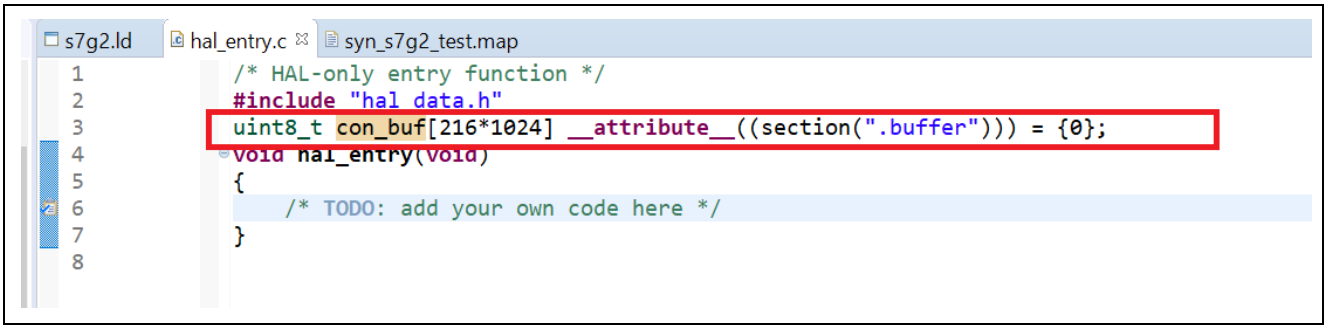


Figure 11. Place Buffer in Reserved Region

4. Expected output and memory allocation

4.1 .map file with IAR linking

Region BUFFER is placed at 0x20008000 from the application code (`hal_entry`) and a 216-KB space is reserved in the RAM as seen in the `.map` file

```
"P5":                                      0x36000
  P5 s0                          0x20008000  0x36000  <Init block>
    BUFFER               inited  0x20008000  0x36000  hal_entry.o [1]
                       - 0x2003e000  0x36000
```

Figure 12. Buffer .map File

4.2 .map file with GCC linking

The section .buffer of length 216KB is placed in the .reserved_section in memory at address 0x20008000 as seen in the .map file

```
.reserved_section
                0x20008000    0x36000
 *(.buffer)
 .buffer        0x20008000    0x36000 ./src/hal_entry.o
                0x20008000                con_buf
```

Figure 13. .buffer in .reserved_section

5. Notes

1. The workaround mentioned above will decrease the available RAM by 216 KB.
2. This workaround splits the SRAM into two non-contiguous memory sections, but the linker is able to seamlessly allocate and assign data to these two sections without developer intervention.
3. Other workarounds :
   a. Do not use SRAM parity error interrupts (or reset). Set PARIOAD.OAD = 0 and NMIER.RPEEN = 0. These are the default settings for the MCU in the Synergy Software Package (SSP). These settings disable parity error interrupts and resets.
   b. Do not use SRAM0 2000_8000h-2003_DFFFh (216 KB)
      # Normal use space: 424 KB (=640 – 216 KB)
   c. If parity error interrupts or resets are enabled and ICLK is greater than 120 MHz, avoid any read access to the 216-KB SRAM0 address (2000 8000h – 2003 DFFFh) (from CPU, DMAC/DTC, EDMAC, JPEG, DRW, LCDC).
   d. Use an ICLK frequency less than 120MHz (ICLK ≦ 120MHz)

6. References

List of the reference documents and links.

- https://gcc.gnu.org/onlinedocs/gcc-4.8.2/gcc/Variable-Attributes.html
- https://www.iar.com/support/tech-notes/linker/how-do-i-place-a-group-of-functions-or-variables-in-a-specific-section/

Website and Support

Visit the following vanity URLs to learn about key elements of the Synergy Platform, download components and related documentation, and get support.

Synergy Software	www.renesas.com/synergy/software
Synergy Software Package	www.renesas.com/synergy/ssp
Software add-ons	www.renesas.com/synergy/addons
Software glossary	www.renesas.com/synergy/softwareglossary
Development tools	www.renesas.com/synergy/tools
Synergy Hardware	www.renesas.com/synergy/hardware
Microcontrollers	www.renesas.com/synergy/mcus
MCU glossary	www.renesas.com/synergy/mcuglossary
Parametric search	www.renesas.com/synergy/parametric
Kits	www.renesas.com/synergy/kits
Synergy Solutions Gallery	www.renesas.com/synergy/solutionsgallery
Partner projects	www.renesas.com/synergy/partnerprojects
Application projects	www.renesas.com/synergy/applicationprojects
Self-service support resources:	
Documentation	www.renesas.com/synergy/docs
Knowledgebase	www.renesas.com/synergy/knowledgebase
Forums	www.renesas.com/synergy/forum
Training	www.renesas.com/synergy/training
Videos	www.renesas.com/synergy/videos
Chat and web ticket	www.renesas.com/synergy/resourcelibrary

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Dec.8.2020	—	First release of document

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 these 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 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, modification, copying or reverse engineering.
5. Renesas Electronics products are classified according to the following two quality grades: "Standard" and "High Quality". The intended applications for each Renesas Electronics product depends 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 system; undersea 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.
6. 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.
7. 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.
8. 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.
9. 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.
10. 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.
11. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. 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.4.0-1 November 2017)

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.

Contact information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit:
www.renesas.com/contact/.