AN12423

SDRAM Code Execution in LPC5460x

Rev. 0 — 20 April 2019 Application Note

Contents

1 Introduction

The article explains the means for code execution in synchronous dynamic random-access memory(SDRAM), and the performance benchmark on LPC5460x. In embedded system designs, SDRAM is used frequently as the memory foot print becomes too large to fit in the on-chip flash/RAM. Executing program in SDRAM is one typical usage of SDRAM.

1.1 SDRAM

The SDRAM used in microcontroller application is single data rate, which can accept one command and transfer one word of data per clock cycle. Typically, in microcontroller applications, SDRAM clock frequencies are 100 MHz or 133 MHz.

1.2 External Memory Controller

In LPC5460x, there is an External Memory Controller(EMC) that supports external memory accessing. It can be configured for supporting different SDRAM.

2 Use SDRAM

2.1 SDRAM initialization

When system is booting, SDRAM is yet initialized, it must be configured before using. In SDRAM initialization functions, SDRAM clock frequency, endianness, and timing parameters should be configured in EMC registers.

SDRAM initialization can be done whenever after booting, however, there still some notes when developer prefers to initialize SDRAM at early stage of booting.

1. SDRAM initialization may use STACK or global variables, developer must make sure the physical memory where STACK or global variables locate is powered.
2. Global variables are not initialized or zeroed before main function starts, developer must set relevant data manually, if SDRAM initialization is performed before main function.

In file board.c, function “BOARD_InitSDRAM” is used to initialize SDRAM, this function can be called in function “SystemInit”, which is invoked by Reset_Handler, the entry of program. As shown in below code snippet, in Reset_Handler, AHBCLKCTRLSET0 register is set to 0x38 to enable SRAM1/SRAM2/SRAM3 clock, if program STACK or SDRAM-related variables is placed in those SRAM regions.

```
Reset_Handler   PROC
EXPORT  Reset_Handler               [WEAK]
```

```
IMPORT  SystemInit
IMPORT  __main
; clock control SRAM1/SRAM2/SRAM3 for stack
LDR     r0, = 0x40000220
MOV     r1, #0x38
STR     r1, [r0]
LDR     r0, =SystemInit
BLX     r0
LDR     r0, =__main
BX      r0
ENDP
```

2.2 Memory allocation

SDRAM can start at four possible address considering that SDRAM chip select has four options, as described in Table 1. SDRAM memory address on page 2.

Table 1. SDRAM memory address

SDRAM Address Range	SDRAM Chip Select
0xA0000000 – 0xA7FFFFFF	SDRAM Chip Select 0 (Up to 256 MB)
0xA8000000 – 0xAFFFFFFF	SDRAM Chip Select 1 (Up to 256 MB)
0xB0000000 – 0xB7FFFFFF	SDRAM Chip Select 2 (Up to 256 MB)
0xB8000000 – 0xBFFFFFFF	SDRAM Chip Select 3 (Up to 256 MB)

Developers can use linker script to allocate code or data into SDRAM, and linker script programming are different among IDEs.

Taking KEIL environment, for example, in below linker script

1. SDRAM region is defined, starts at 0xA0000000, and size is 0x80000.
2. Data or code with “SDRAM_Data” and “SDRAM_Function” attributes are placed in SDRAM region.
3. Data and code in “core_list_join.c”, “core_matrix.c”, “core_state.c” and “core_util.c” are placed in SDRAM region.

```
#define m_sdram_start      0xA0000000
#define m_sdram_size       0x80000
LR_m_sdram_text   m_sdram_start    m_sdram_size {
    *(SDRAM_Data)
    *(SDRAM_Function)
    core_list_join.o
    core_matrix.o
    core_state.o
    core_util.o
}
```

2.3 MPU configuration

As mentioned in Cortex™-M4 Devices Generic User Guide, the default configuration of memory 0xA0000000 – 0xDFFFFFFF is not executable. When code in SDRAM is executed with MPU disable, default configuration takes effect, Arm CM4 should issue

hard fault event with instruction access violation flag, where SCB->CFSR has bit 1 set See Figure 1. on page 3. The fault indicates that the processor attempted an instruction fetch from a location that does not permit execution.

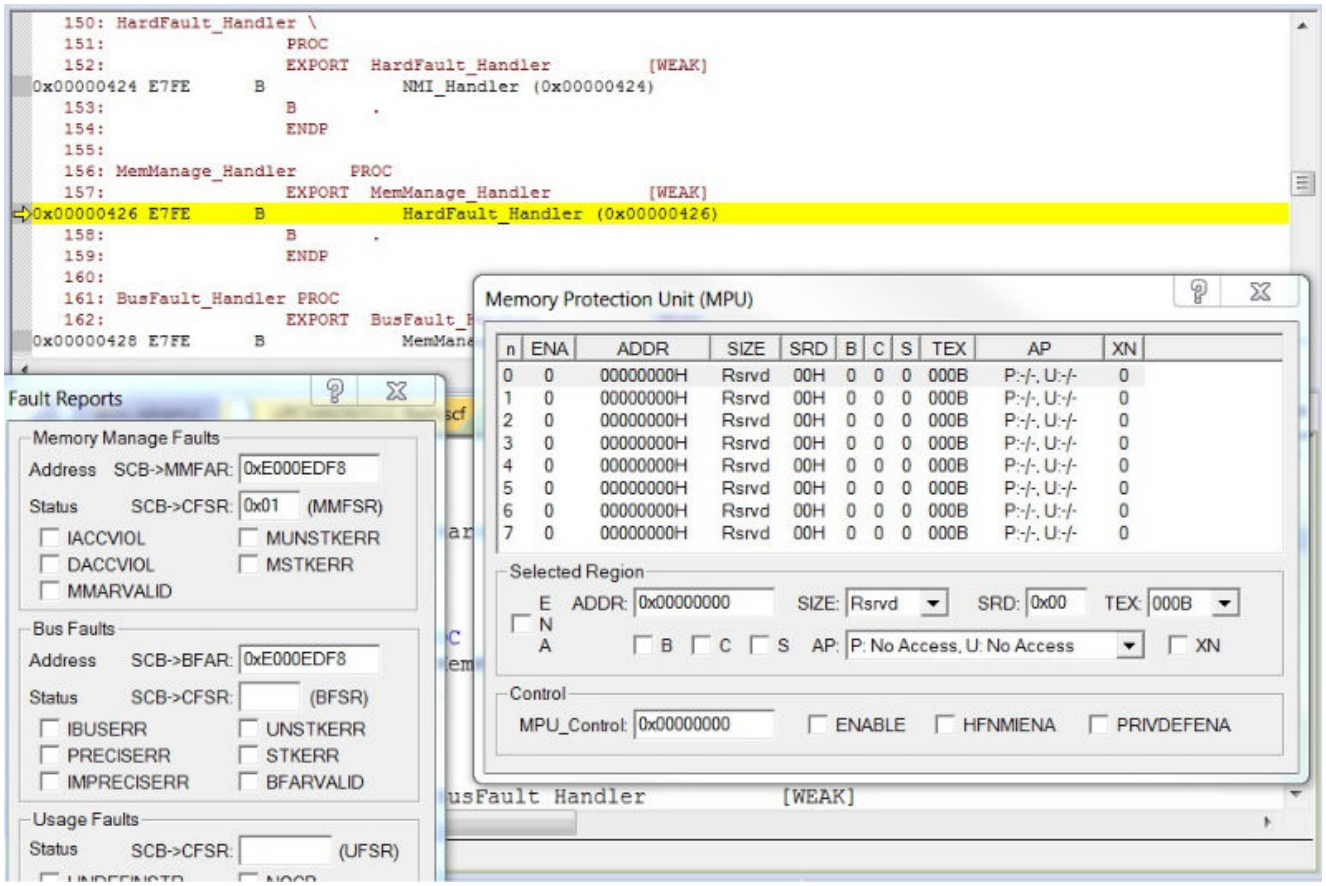

Figure 1. IACCVIOL hard fault

Thus, if developer would execute code in SDRAM, MPU configuration must be changed. In below code snippet, MPU is configured which allows memory region from 0xA0000000 to 0xA0100000 is executable.

MPU->RNR = 0; //Region number 0

MPU->RBAR = 0xA0000000;//Region base address

/ Full Access | TEX: 000 | S: 0 | C: 0 | B:0 (No cacheable, no shareable)| 1M SIZE | ENABLE */*

MPU->RASR = (0 << 28) | (0x3 << 24) | (0x0 << 19) | (0 << 18) | (0 << 17) | (0 << 16) | (0xFF < 8) | (0x13 << 1) | (1 << 0); //Region size and enable

MPU->CTRL = MPU_CTRL_ENABLE_Msk | MPU_CTRL_PRIVDEFENA_Msk;

3 SDRAM performance benchmark

SDRAM code execution performance benchmark is measured using CoreMark, as shown in Table 2. SDRAM Performance benchmark using CoreMark on page 4. From the measurement, we may conclude that:

1. Code execution in SDRAM has only 40 % performance against running code in SRAMX (ICODE accessible) SRAM and ~50 % performance against running code in on-chip FLASH.
2. Higher CPU frequency has no help when code is running in SDRAM, SDRAM bandwidth is bottleneck

Table 2. SDRAM Performance benchmark using CoreMark

Basic Information CoreMark Size: 666 Iterations: 4000 Compiler version: Arm C/C++ Compiler, 5.24.2 [Build 528] Compiler flags: --c99 -c --cpu Cortex-M4.fp -D__MICROLIB -g -O3 -Otime --apcs=interwork Memory location: STATIC	
Memory Configuration	**Time per Iteration (ms)**
Code executing in SRAMX, Arm CM4@96 MHz	4.015
Code executing in SRAMX, Arm CM4@180 MHz	2.141
Code executing in On-Chip FLASH, Arm CM4@96 MHz	4.869
Code executing in On-Chip FLASH, Arm CM4@180 MHz	3.228
Code executing in SDRAM@96MHz, Arm CM4@96 MHz	10.050
Code executing in SDRAM@180MHz, Arm CM4@90 MHz	10.458

4 Reference documents

Source	Title
NXP Semiconductors	LPC5460x data sheet
NXP Semiconductors	LPC5460x user manual
NXP Semiconductors	OM13090 user manual
ARM Ltd.	Cortex-M4_ReferenceManual
ARM Ltd.	Cortex-M4 Technical Reference Manual Revision r0p1

5 Revision history

Table 3. Revision history

Rev. No.	Date	Description
0	20 April 2019	Initial release

How To Reach Us

Home Page:

nxp.com

Web Support:

nxp.com/support

Information in this document is provided solely to enable system and software implementers to use NXP products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document. NXP reserves the right to make changes without further notice to any products herein.

NXP makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does NXP assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. “Typical” parameters that may be provided in NXP data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including “typicals,” must be validated for each customer application by customer's technical experts. NXP does not convey any license under its patent rights nor the rights of others. NXP sells products pursuant to standard terms and conditions of sale, which can be found at the following address: nxp.com/SalesTermsandConditions.

While NXP has implemented advanced security features, all products may be subject to unidentified vulnerabilities. Customers are responsible for the design and operation of their applications and products to reduce the effect of these vulnerabilities on customer’s applications and products, and NXP accepts no liability for any vulnerability that is discovered. Customers should implement appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP, the NXP logo, NXP SECURE CONNECTIONS FOR A SMARTER WORLD, COOLFLUX, EMBRACE, GREENCHIP, HITAG, I2C BUS, ICODE, JCOP, LIFE VIBES, MIFARE, MIFARE CLASSIC, MIFARE DESFire, MIFARE PLUS, MIFARE FLEX, MANTIS, MIFARE ULTRALIGHT, MIFARE4MOBILE, MIGLO, NTAG, ROADLINK, SMARTLX, SMARTMX, STARPLUG, TOPFET, TRENCHMOS, UCODE, Freescale, the Freescale logo, AltiVec, C-5, CodeTEST, CodeWarrior, ColdFire, ColdFire+, C-Ware, the Energy Efficient Solutions logo, Kinetis, Layerscape, MagniV, mobileGT, PEG, PowerQUICC, Processor Expert, QorIQ, QorIQ Qonverge, Ready Play, SafeAssure, the SafeAssure logo, StarCore, Symphony, VortiQa, Vybrid, Airfast, BeeKit, BeeStack, CoreNet, Flexis, MXC, Platform in a Package, QUICC Engine, SMARTMOS, Tower, TurboLink, and UMEMS are trademarks of NXP B.V. All other product or service names are the property of their respective owners. AMBA, Arm, Arm7, Arm7TDMI, Arm9, Arm11, Artisan, big.LITTLE, Cordio, CoreLink, CoreSight, Cortex, DesignStart, DynamIQ, Jazelle, Keil, Mali, Mbed, Mbed Enabled, NEON, POP, RealView, SecurCore, Socrates, Thumb, TrustZone, ULINK, ULINK2, ULINK-ME, ULINK-PLUS, ULINKpro, µVision, Versatile are trademarks or registered trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere. The related technology may be protected by any or all of patents, copyrights, designs and trade secrets. All rights reserved. Oracle and Java are registered trademarks of Oracle and/or its affiliates. The Power Architecture and Power.org word marks and the Power and Power.org logos and related marks are trademarks and service marks licensed by Power.org.

For more information, please visit: http://www.nxp.com
For sales office addresses, please send an email to: salesaddresses@nxp.com

Date of release: 20 April 2019
Document identifier: AN12423

arm