TN00037

LPC54018 Code Partition between SRAM and SPIFI memory

1.0 — 14 March 2018 **Technical note**

Document information

Info	Content
Keywords	LPC540xx, SRAM, SPIFI, XIP, linker scripts
Abstract	This technical note describes how to partition code between SRAM and SPIFI memory regions using linker scripts.

Revision history

Rev	Date	Description
1.0	20180314	Initial version.

Contact information

For more information, please visit: http://www.nxp.com

For sales office addresses, please send an email to: salesaddresses@nxp.com

1. Introduction

The LPC540xx is a family of ARM Cortex-M4 based microcontrollers used in embedded applications. This technical note references the LPCXpresso development board for LPC540xx MCUs. For details of the board, see:

https://www.nxp.com/demoboard/om40003

Fig 1. LPC54018 LPCXpresso development board

2. Description

The "blinky_sram_xip_hybrid" example illustrates how to split code between SRAMX and SPIFI memory regions. The example is a simple blinky application toggling LED3 on the LPC54018 LPCXpresso development board. The delay() function in “delay_nop.c” file is placed in SPIFI memory location 0x10040000 and is executed-in-place (XIP). The rest of the code is linked to SRAMX memory region and is executed in plain load fashion.

Note: See the “Getting Started with MCUXpresso SDK for LPC540xx.pdf” document in “docs” folder of SDK package. This document explains how to configure the IDEs for various debug configurations (SRAMX, Plain Load, and XIP).

The example is available in two tool chains:

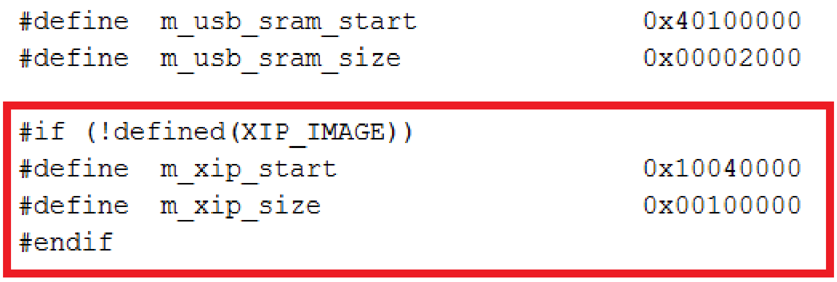
- IAR embedded Workbench
- Keil MDK

The Keil and IAR examples are found in:

lpc54018_blinky_sram_xip_hybrid_keil_iar\boards\lpcxpresso54018\demo_apps\blinky_sram_xip_hybrid

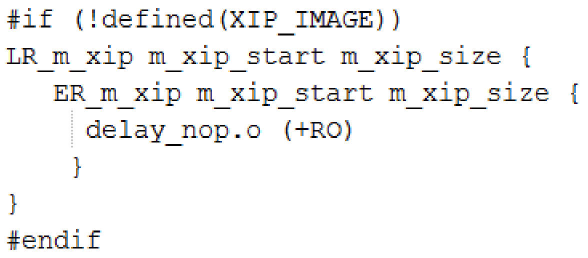
2.1 Keil MDK IDE

The Keil linker script modification is shown below. Refer to “LPC54018_sram_spifi_flash_hybrid.scf” file in the example. An XIP region is defined and the “delay_nop.o” object file is placed in this defined region.



```
#define  m_usb_sram_start              0x40100000
#define  m_usb_sram_size               0x00002000

#if (!defined(XIP_IMAGE))
#define  m_xip_start                   0x10040000
#define  m_xip_size                    0x00100000
#endif
```

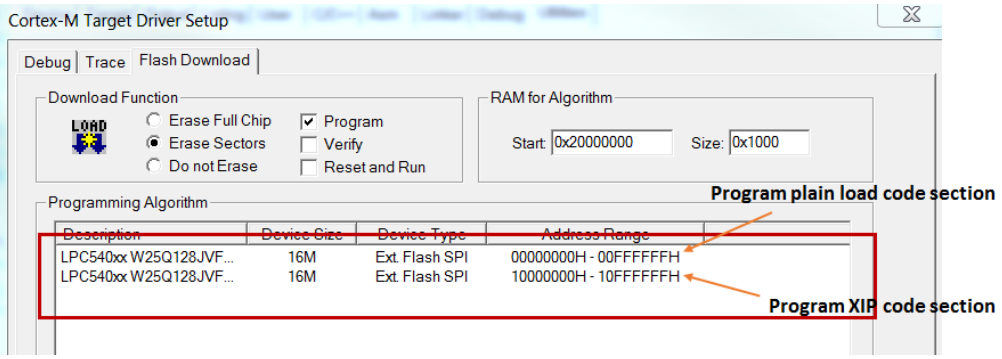
Fig 2. Keil linker script – definition memory region



```
#if (!defined(XIP_IMAGE))
LR_m_xip m_xip_start m_xip_size {
   ER_m_xip m_xip_start m_xip_size {
     delay_nop.o (+RO)
    }
}
#endif
```

Fig 3. Keil linker script – placement of delay_nop.o

To program the plain load example with the XIP code region, add the Quad SPI flash flashloaders. See Fig 4. One flashloader has address range from 0x00000000 – 0x00FFFFFF to program Plain load image and another flashloader has the address range 0x10000000 – 0x10FFFFFF to program the XIP code section.



Download Function
○ Erase Full Chip
● Erase Sectors
○ Do not Erase
☑ Program
☐ Verify
☐ Reset and Run

RAM for Algorithm
Start: 0x20000000 Size: 0x1000

Programming Algorithm

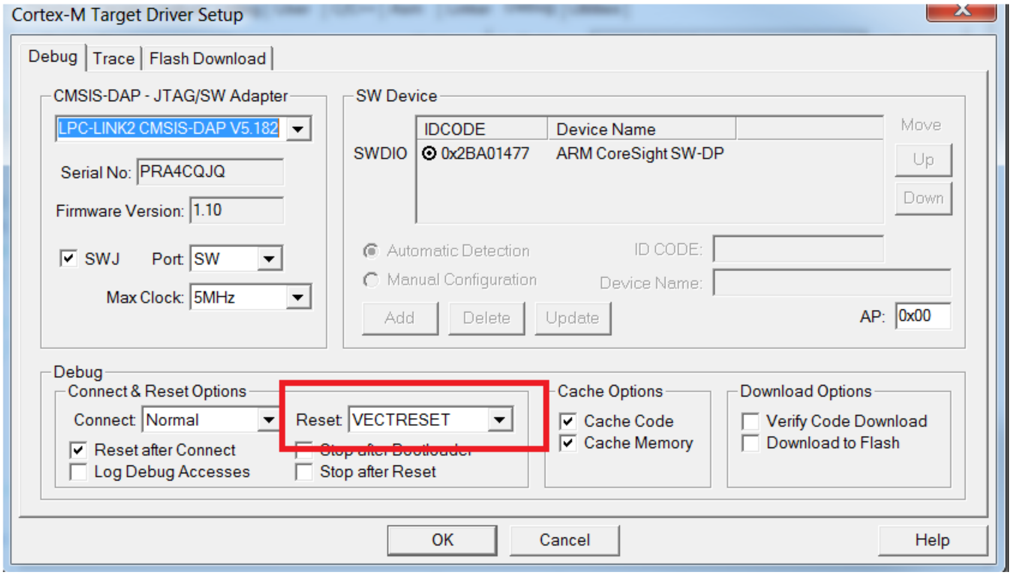
Description	Device Size	Device Type	Address Range
LPC540xx W25Q128JVF...	16M	Ext. Flash SPI	00000000H - 00FFFFFFH
LPC540xx W25Q128JVF...	16M	Ext. Flash SPI	10000000H - 10FFFFFFH

Program plain load code section

Program XIP code section

Fig 4. Keil flashloaders

In Keil MDK, to program the application into external Quad SPI flash device, set the Reset type in Keil IDE to SYSRESETREQ (See Project->Options->Debug tab ->Settings). To debug the application set the Reset type to VECTRESET.



CMSIS-DAP - JTAG/SW Adapter
LPC-LINK2 CMSIS-DAP V5.182
Serial No: PRA4CQJQ
Firmware Version: 1.10
☑ SWJ Port: SW
Max Clock: 5MHz

SW Device

	IDCODE	Device Name
SWDIO	0x2BA01477	ARM CoreSight SW-DP

● Automatic Detection ID CODE:
○ Manual Configuration Device Name:
AP: 0x00

Debug
Connect & Reset Options
Connect: Normal
Reset: VECTRESET
☑ Reset after Connect
☐ Stop after Bootloader
☐ Log Debug Accesses
☐ Stop after Reset

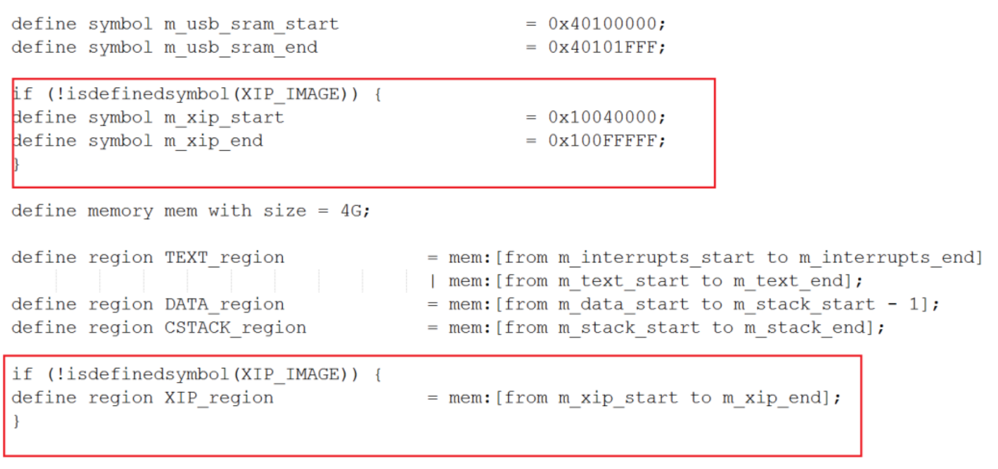
Cache Options
☑ Cache Code
☑ Cache Memory

Download Options
☐ Verify Code Download
☐ Download to Flash

Fig 5. Keil Debugger Reset Type

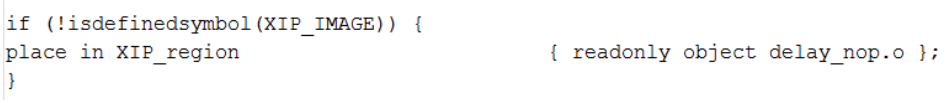
2.2 IAR Embedded Workbench IDE

The IAR linker script modification is shown below. See the “LPC54018_sram_spifi_flash_hybrid.icf” file in the example.



```
define symbol m_usb_sram_start                   = 0x40100000;
define symbol m_usb_sram_end                     = 0x40101FFF;

if (!isdefinedsymbol(XIP_IMAGE)) {
define symbol m_xip_start                        = 0x10040000;
define symbol m_xip_end                          = 0x100FFFFF;
}

define memory mem with size = 4G;

define region TEXT_region               = mem:[from m_interrupts_start to m_interrupts_end]
                                        | mem:[from m_text_start to m_text_end];
define region DATA_region               = mem:[from m_data_start to m_stack_start - 1];
define region CSTACK_region             = mem:[from m_stack_start to m_stack_end];

if (!isdefinedsymbol(XIP_IMAGE)) {
define region XIP_region                = mem:[from m_xip_start to m_xip_end];
}
```

Fig 6. IAR linker script – definition of memory region



```
if (!isdefinedsymbol(XIP_IMAGE)) {
place in XIP_region                       { readonly object delay_nop.o };
}
```

Fig 7. IAR linker script - placement of delay_nop.o

In IAR IDE, no changes to the flashloaders are required because the IAR in-built flashloaders take care of programming both the plain load and XIP sections.

The MCUXpresso IDE version of example will be released with the next release of MCUXpresso IDE.

3. Legal information

3.1 Definitions

3.2 Disclaimers

3.3 Trademarks

4. Contents

Please be aware that important notices concerning this document and the product(s) described herein, have been included in the section 'Legal information'.

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

Date of release: 14 March 2018
Document identifier: TN00037