NXP Semiconductors
Application Note

Document Number: AN12279
Rev. 0, 11/2018

Configure QSPI Bus Width and Frequency in Pre-Boot Loader Stage on QorIQ LS Series Processors

Contents

1 Introduction

When QSPI is selected as the boot source, PBL accesses QSPI flash in single bit mode and fetches RCW/PBI at a low speed. If the boot loader is also in the QSPI flash, the boot loader instructions will also be fetched at the same speed. This slows down the boot process.

By increasing QSPI clock frequency and expanding QSPI bus width at an early stage in boot process, the boot process can be sped up.

This application note is intended for users who want to optimize boot time in their applications when QSPI is selected as the boot source. The purpose of this document is to allow users to reconfigure the QSPI mode in pre-boot loader stage to operate at higher frequency and wider bus width.

2 Pre-boot process

The table below shows cfg_rcw_src values for different QorIQ LS series processors to select QSPI as the boot source.

Table 1. cfg_rcw_src values for LS series processors

SoC	cfg_rcw_src value (in binary)
LS1012A	1
LS1021A	0_0100_010x
LS1043A	
LS1046A	

3 Increasing QSPI bus frequency

QSPI bus frequency can be increased by writing to SCFG_QSPI_CFG register in PBI. For example, in below PBI command, base frequency is divided by 32 to achieve higher QSPI bus frequency with SCFG_QSPI_CFG[CLK_SEL] = 4'b0010:

```
0957015C 40100000
```

QSPI bus frequency can be modified at any stage, but it must not exceed the maximum QSPI bus frequency mentioned in the SoC data sheet.

4 Increasing QSPI bus width

QSPI bus width should be increased when QSPI bus is not in use. A small firmware can be used to change QSPI bus width. The firmware can be made to execute from OCRAM. The firmware can be loaded to OCRAM using PBI commands. SCFG_SCRATCHRW1/2 registers can be written to make Core 0 execute the firmware in OCRAM. The firmware finally jumps to the actual boot loader residing in QSPI. All further instructions fetched from QSPI will be with increased bus width.

For increasing QSPI bus width, minimal firmware is required that can perform the following functions:

- Unlock QSPI LUTs so that they can be programmed
- Update QSPI LUTs to change QSPI mode
- Lock QSPI LUTs to protect them from writing
- Jump to U-Boot code on QSPI

The minimal firmware changes QSPI mode by configuring LUTs with the required QSPI mode commands and number of QSPI pads to be used. PBL programs OCRAM with minimal firmware for executing above functions.

QCVS tool of CodeWarrior for Armv8 is used for modifying PBL file. See *QCVS PBL Tool User Guide* to know how to work with the PBL configuration tool.

4.1 Unlock LUTs

Perform the following steps to unlock QSPI LUTs:

1. Program QuadSPI_LUTKEY at 0x1550300 as 0x5AF05AF0.
2. Set QuadSPI_LCKCR[UNLOCK] at 0x1550304.

4.2 Update LUTs for Dual-Bit mode

Perform the following steps to update QSPI LUTs for Dual-Bit mode:

1. Update LUT0 at 0x1550310 – 0x091804bb.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
INSTR1						PAD1		OPRND1							
0	0	0	0	1	0	0	1	0	0	0	1	1	0	0	0

16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
INSTR0						PAD0		OPRND0							
0	0	0	0	0	1	0	0	1	0	1	1	1	0	1	1

Field	Description	Value
INSTR1	ADDR	0b000010
PAD1	Two pads	0b01
OPRND1	24 address bits	0b00011000
INSTR0	CMD	0b000001
PAD0	One pad	0b00
OPRND0	Dual I/O read	0b10111011[1]

1. This vaue is meant for S25FS128S Cypress QSPI flash. For any other flash device, see data sheet of the flash device.

2. Update LUT1 at 0x1550314 – 0x1d080d0c.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
INSTR1						PAD1		OPRND1							
0	0	0	1	1	1	0	1	0	0	0	0	1	0	0	0

16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
INSTR0						PAD0		OPRND0							
0	0	0	0	1	1	0	1	0	0	0	0	1	1	0	0

Field	Description	Value
INSTR1	READ	0b000111
PAD1	Two pads	0b01
OPRND1	8 bytes read data size	0b00001000

Table continues on the next page...

Increasing QSPI bus width

Field	Description	Value
INSTR0	DUMMY	0b000011
PAD0	Two pads	0b01
OPRND0	12 dummy clock cycles[1]	0b00001100

1. The number of dummy clock cycles is determined by the latency code in the configuration register (CR2V[3:0]) of QSPI flash. See data sheet of the flash device for more details.

3. Update LUT2 at 0x1550318 – 0x00002400.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
INSTR1						PAD1		OPRND1							
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
INSTR0						PAD0		OPRND0							
0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0

Field	Description	Value
INSTR1		0b000000
PAD1		0b00
OPRND1		0b00000000
INSTR0	JUMP_ON_CS	0b001001
PAD0	One pad	0b00
OPRND0	Jumps to instruction number 0 when CS is deasserted	0b00000000

4.3 Update LUTs for Quad-Bit mode

Perform the following steps to update QSPI LUTs for Quad-Bit mode:

1. Update LUT0 at 0x1550310 – 0x0a1804eb.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
INSTR1						PAD1		OPRND1							
0	0	0	0	1	0	1	0	0	0	0	1	1	0	0	0

16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
INSTR0						PAD0		OPRND0							
0	0	0	0	0	1	0	0	1	1	1	0	1	0	1	1

Field	Description	Value
INSTR1	ADDR	0b000010
PAD1	Four pads	0b10
OPRND1	24 address bits	0b00011000
INSTR0	CMD	0b000001
PAD0	One pad	0b00
OPRND0	Quad I/O read	0b11101011[1]

1. This vaue is meant for S25FS128S Cypress QSPI flash. For any other flash device, see data sheet of the flash device.

2. Update LUT1 at 0x1550314 – 0x1e080e0a.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
INSTR1						PAD1		OPRND1							
0	0	0	1	1	1	1	0	0	0	0	0	1	0	0	0

16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
INSTR0						PAD0		OPRND0							
0	0	0	0	1	1	1	0	0	0	0	0	1	0	1	0

Field	Description	Value
INSTR1	READ	0b000111
PAD1	Four pads	0b10
OPRND1	8 bytes read data size	0b00001000
INSTR0	DUMMY	0b000011
PAD0	Four pads	0b10
OPRND0	12 dummy clock cycles[1]	0b00001010

1. The number of dummy clock cycles is determined by the latency code in the configuration register (CR2V[3:0]) of QSPI flash. See data sheet of the flash device for more details.

3. Update LUT2 at 0x1550318 – 0x00002400.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
INSTR1						PAD1		OPRND1							
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
INSTR0						PAD0		OPRND0							

Table continues on the next page...

Increasing QSPI bus width

0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0

Field	Description	Value
INSTR1		0b000000
PAD1		0b00
OPRND1		0b00000000
INSTR0	JUMP_ON_CS	0b001001
PAD0	One pad	0b00
OPRND0	Jumps to instruction number 0 when CS is deasserted	0b00000000

4.4 Lock LUTs

Perform the following steps to lock QSPI LUTs:

1. Program QuadSPI_LUTKEY at 0x1550300 as 0x5AF05AF0.
2. Set QuadSPI_LCKCR[LOCK] at 0x1550304.

4.5 Jump to boot loader

Jump to 0x40100000 address of QSPI where actual boot loader resides.

4.6 Assembly code for Dual-Bit mode

Below is the assembly code to increase QSPI bus width by switching QSPI to Dual-Bit mode:

```
//Unlock LUTs
ldr x2,=0x1550300
ldr w3,=0xf05af05a
str w3,[x2]
ldr x2,=0x1550304
ldr w3,=0x02000000
str w3,[x2]
//Update LUT0
ldr x2,=0x1550310
ldr w3,=0xbb041809
str w3,[x2]
//Update LUT1
ldr x2,=0x1550314
ldr w3,=0x0c0d081d
str w3,[x2]
//Update LUT2
ldr x2,=0x1550318
ldr w3,=0x00240000
str w3,[x2]
//Lock LUTs
ldr x2,=0x1550304
ldr w3,=0x01000000
```

```
str w3,[x2]
//Jump to U-Boot of QSPI
mov x30,#0x40100000
br x30
```

4.7 Assembly code for Quad-Bit mode

Below is the assembly code to increase QSPI bus width by switching QSPI to Quad-Bit mode:

```
//Unlock LUTs
ldr x2,=0x1550300
ldr w3,=0xf05af05a
str w3,[x2]
ldr x2,=0x1550304
ldr w3,=0x02000000
str w3,[x2]
//Update LUT0
ldr x2,=0x1550310
ldr w3,=0xeb04180a
str w3,[x2]
//Update LUT1
ldr x2,=0x1550314
ldr w3,=0x0a0e081e
str w3,[x2]
//Update LUT2
ldr x2,=0x1550318
ldr w3,=0x00240000
str w3,[x2]
//Lock LUTs
ldr x2,=0x1550304
ldr w3,=0x01000000
str w3,[x2]
//Jump to U-Boot of QSPI
mov x30,#0x40100000
br x30
```

4.8 Generate binary file of SPL code

The SPL code is in assembly language. Follow below steps to create binary file of SPL code in Ubuntu:

1. Make object (.o) file from assembly (.S) file:

   ```
   aarch64-linux-gnu-gcc -c <assembly_filename.S>
   ```
2. Make binary (.bin) file from object (.o) file:

   ```
   aarch64-linux-gnu-objcopy -O binary <object_filename.o>
   <binary_filename.bin>
   ```

5 PBL configuration using CodeWarrior QCVS tool

It is recommended to use CodeWarrior QCVS tool to generate the PBI image. See *QCVS PBL Tool User Guide* to know how to work with the PBL configuration tool, for example, how to create a new QorIQ configuration project, how to configure the PBL component, and how to perform basic operations using the PBL configuration tool, such as generating RCW+PBI binary file.

Following changes are required in PBI commands using QCVS tool:

1. Write 0x10000000 (the location of OCRAM) on the SCFG_SCRATCHRW2 register (at 0x1570604) and 0x00000000 on the SCFG_SCRATCHRW1 register (at 0x1570600), using the following PBI commands:

```
09570600 00000000
09570604 00000000
```

2. Program OCRAM with minimal firmware using the following PBI commands:

```
09570158 00001000
81000000 <64 bytes binary data>
81000040 <64 bytes binary data>
…
```

NOTE

Zeros must be added at the end of binary file of minimal firmware to make the file size a multiple of 128 bytes.

6 Summary

The QSPI bus frequency can be increased by writing to a register in PBI. For increasing QSPI bus width, example given in this document can be used to load a firmware on OCRAM. The firmware changes LUTs accordingly and jumps to the existing boot loader code in QSPI flash. Increasing frequency and bus width can help optimize boot time when QSPI flash is the boot source.

7 Revision history

The table below summarizes revisions to this document.

Table 2. Revision history

Revision	Date	Topic cross-reference	Change description
Rev. 0	11/2018		Initial public release

How to Reach Us:

Home Page:
nxp.com

Web Support:
nxp.com/support

Document Number AN12279
Revision 0, 11/2018