NXP Semiconductors
Application Note

Document Number: AN12098
Rev. 0, 12/2017

LS1043A Interfacing with CPLD in LS1043ARDB

Contents

1 Introduction

A QorIQ LS series processor can be connected with a CPLD/FPGA device through various interfaces, such as I2C, IFC, USB, SPI, PCIe, TDM, RGMII, and so on. This document explains how IFC interface is used to connect the LS1043A processor and CPLD used on the LS1043A reference design board (RDB). The concept can be used to connect any QorIQ family device having IFC to FPGA/CPLD on any custom board.

2 Acronyms and abbreviations

The table below lists and describes the acronyms and abbreviations used in this document.

Table 1. Acronyms and abbreviations

Acronym/abbreviation	Description
CPLD	Complex programmable logic device
CS	Chip select
FPGA	Field-programmable gate array

Table continues on the next page...

Table 1. Acronyms and abbreviations (continued)

Acronym/abbreviation	Description
GPCM	General-purpose chip-select machine
IFC	Integrated flash controller
SoC	System-on-chip

3 Recommended resources

The table below lists the additional documents and tools you are recommended to use.

Table 2. Recommended documents and tools

Document	Description	Location / how to get
Recommended documents		
LS1043A Chip Errata (LS1043ACE)	This document describes the latest fixes and workarounds for the LS1043A SoC. You are strongly recommended to read this document thoroughly before starting a new product design with the SoC.	Contact your NXP representative
QorIQ LS1043A, LS1023A Data Sheet (LS1043A)	Contains information on LS1043A/LS1023A pin assignments, electrical characteristics, hardware design considerations, package information, and ordering information	www.nxp.com
QorIQ LS1043A Reference Manual (LS1043ARM)	Provides a detailed description on the LS1043A multicore processor and its features, such as memory map, serial interfaces, power supply, chip features, and clock information	www.nxp.com
QorIQ LS1043A Product Brief (LS1043APB)	Briefly explains the LS1043A processor, its features, and application examples	www.nxp.com
QorIQ LS1043ARDB Getting Started Guide (LS1043ARDBGSG)	Describes the LS1043ARDB hardware kit, provides settings for the onboard switches, connectors, jumpers, and LEDs, and explains the basic board operations in a step-by-step manner	www.nxp.com
QorIQ LS1043A Reference Design Board Reference Manual (LS1043ARDBRM)	Explains the LS1043ARDB interfaces and configuration	www.nxp.com
LS1043ARDB schematics	Provides circuitry details about the LS1043ARDB	www.nxp.com
Software-enablement tools		
QorIQ LS1043A reference design board	A high-performance reference design board based on the QorIQ LS1043A processor	www.nxp.com

4 LS1043ARDB block diagram

The figure below shows the block diagram of the LS1043ARDB. The IFC and CPLD components are highlighted only to indicate that this document is based on the communication between these two components.

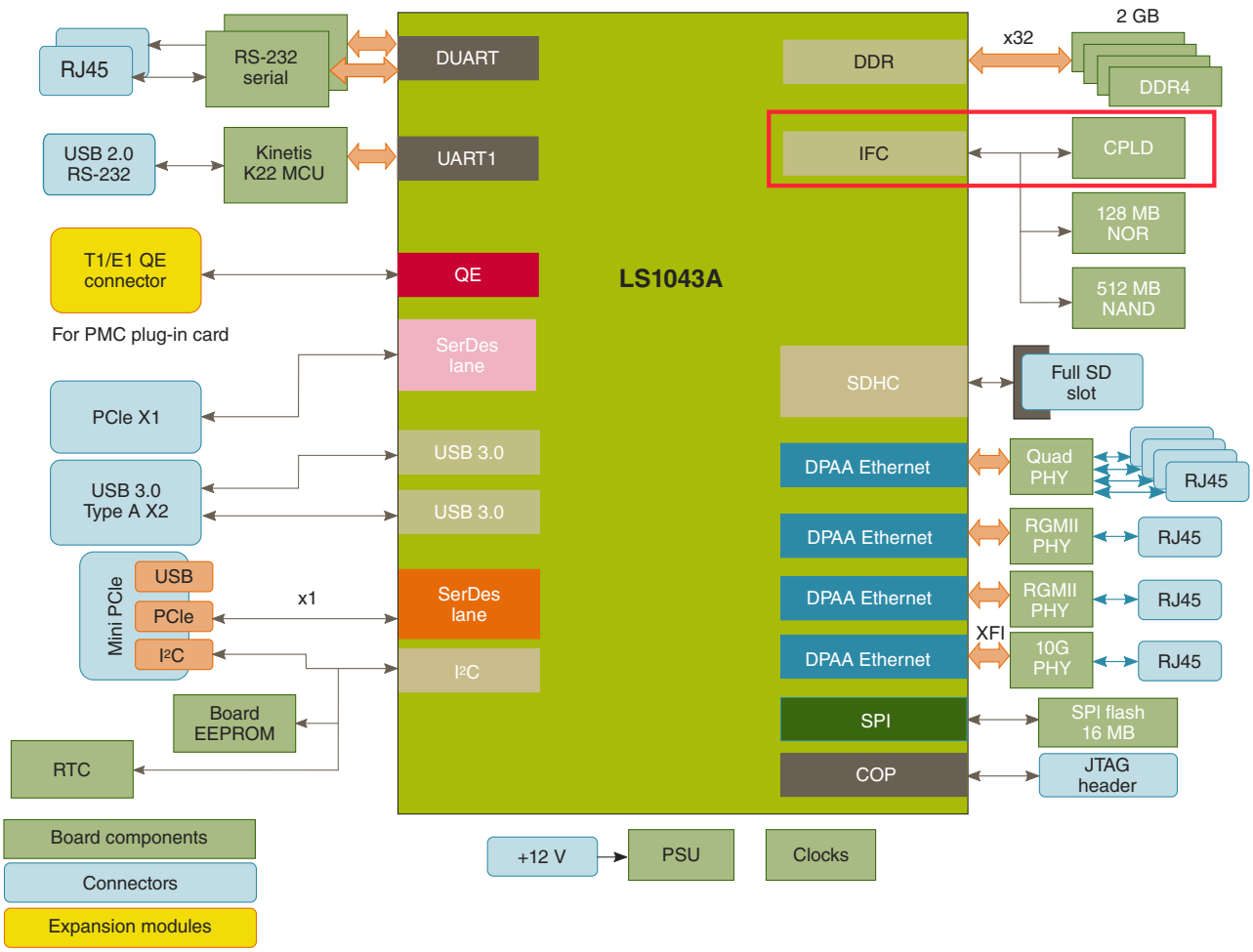

Figure 1. LS1043ARDB block diagram

5 LS1043ARDB CPLD overview

The LS1043ARDB has an onboard CPLD (part number: LCMXO1200C-3FTN256C) from Lattice Semiconductor.

In the LS1043ARDB, the CPLD implements registers that are mapped to the board control and status register (BCSR) memory. The LS1043A processor can access these registers through its IFC interface.

The table below shows the CPLD memory details, including peripheral data bus width.

Table 3. CPLD memory

Address range (hex)	Chip select	Bank size	Data width	Access
0x7FB00000 - 0x7FB0FFFF	CS2	64 KB	8 bits	Read/write

The table below shows the memory map for the CPLD registers.

Table 4. CPLD registers

Offset address (hex)	Register name	Data width (bits)	Access
0	CPLD major version register (CPLD_VER)	8	R
1	CPLD minor version register (CPLD_VER_SUB)	8	R
2	PCBA version register (CPLD_PCBA_VER)	8	R
3	System reset register (CPLD_SYSTEM_RST)	8	R/W
4	CPLD override physical switches enable register (CPLD_SOFT_MUX_ON)	8	R/W
5	POR RCW source location register 1 (CPLD_REG_RCW_SRC1)	8	R/W
6	POR RCW source location register 2 (CPLD_REG_RCW_SRC2)	8	R/W
7	Flash bank selection register (CPLD_REG_BANK)	8	R/W
8	System clock single-ended or differential input selection register (CPLD_SYSCLK_SEL)	8	R/W
9	UART1 output selection register (CPLD_UART_SEL)	8	R/W
A	SerDes PLL1 reference clock input selection register (CPLD_SD1REFCLK_SEL)	8	R/W
B	TDM clock or SDHC/USB selection register (CPLD_TDMCLK_MUX_SEL)	8	R/W
D	Status LED control register (CPLD_STATUS_LED)	8	R/W
E	Global reset register (CPLD_GLOBAL_RST)	8	R/W
F	TDM riser card presence detection register (CPLD_TDMR_PRS_N)	8	R
10	RTC clock assignment register (CPLD_REG_RTC)	8	R/W
11	EVDD control register (CPLD_EVDD_SEL)	8	R/W
12	CPLD register override physical switch SDHC_VS/ SPI_CS0 enable register (CPLD_SOFT_VS_SPICS0)	8	R/W
13	SDHC_VS or SPI_CS0 selection register (CPLD_VS_SPICS0_SEL)	8	R/W

For more details on these registers, see *QorIQ LS1043A Reference Design Board Reference Manual* (LS1043ARDBRM).

6 Hardware interfacing between LS1043A IFC and CPLD

The figure below shows how the LS1043A processor and CPLD can communicate using IFC. Here, the CPLD is connected to the LS1043A processor through an 8-bit data bus on chip select 2 (CS2) of IFC. The communication between the two devices is asynchronous.

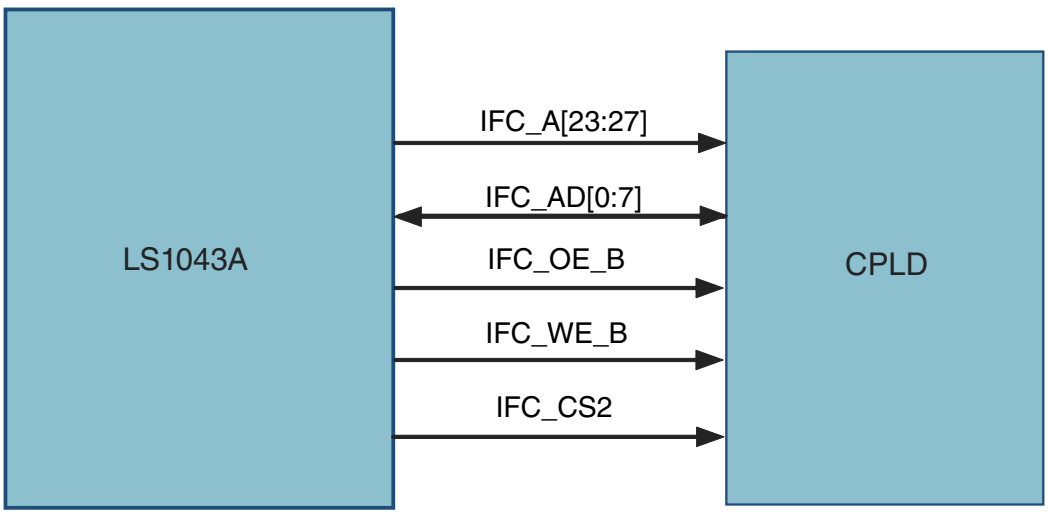

Figure 2. Connection diagram between LS1043A IFC and CPLD

The table below shows a mapping between signals of the LS1043A IFC bus and CPLD bus.

Table 5. Mapping between IFC and CPLD bus signals

Signal description	Signal name	
	IFC bus	CPLD bus
Address bus	IFC_A[23:27]	CPLD_ADDR[4:0]
Data bus	IFC_AD[0:7]	CPLD_DATA[7:0]
Output enable	IFC_OE_B	IFC_OE_B
Write enable	IFC_WE_B	IFC_WE_B
Chip select	IFC_CS2	CPLD_CS

7 LS1043A IFC configuration

The LS1043A integrated flash controller (IFC) is configured in Normal GPCM mode and CPLD is mapped on CS2 of IFC.

The table below shows the details of IFC configuration registers.

Table 6. IFC configuration registers

Register name	Absolute address	Data value
Chip-select Property register 2 (IFC_CSPR2)	0x153_0028	0x8500b07f
Address Mask register 2 (IFC_AMAS2K)	0x153_00B8	0x0000ffff

Table continues on the next page...

Table 6. IFC configuration registers (continued)

Register name	Absolute address	Data value
Flash Timing register 0 for CS2 - Normal GPCM Mode (IFC_FTIM0_CS2_GPCM)	0x153_0220	0x0f000ff0
Flash Timing register 1 for CS2 - Normal GPCM Mode (IFC_FTIM1_CS2_GPCM)	0x153_0224	0x003f00ff
Flash Timing register 2 for CS2 - Normal GPCM Mode (IFC_FTIM2_CS2_GPCM)	0x153_0228	0xff003c0f
Flash Timing register 3 for CS2 - Normal GPCM Mode (IFC_FTIM3_CS2_GPCM)	0x153_022C	0x00000000

The chip-select property register configures IFC interface as follows:

- Base address: 0x7FB0_0000
- Port size: 8 bits
- TE signal active low
- Interface: GPCM
- Bank is valid

The address mask register sets bank size as 64 KB.

Address Range: 0x7FB00000 - 0x7FB0FFFF

8 DC specifications

The table below provides the DC electrical characteristics for the LS1043A IFC interface when operating at OVDD = 1.8 V.

Table 7. LS1043A IFC DC specification

Parameter	Symbol	Min	Max	Unit
Input high voltage	VIH	1.2	-	V
Input low voltage	VIL	-	0.6	V
Output high voltage	VOH	1.6	-	V
Output low voltage	VOL	-	0.32	V

The table below provides the DC electrical characteristics for CPLD (LCMXO1200C-3FTN256C) interface when operating at VCCIO = 1.8 V.

Table 8. CPLD DC specification

Parameter	Symbol	Min	Max	Unit
Input high voltage	VIH	1.17	-	V
Input low voltage	VIL	-	0.63	V
Output high voltage	VOH	1.4	-	V
Output low voltage	VOL	-	0.4	V

For communication between the IFC and CPLD devices, DC specification of each device should match DC specification of its link partner. As shown in the tables above, VOH/L specifications of the LS1043A IFC meet VIH/L requirements of the CPLD and vice-versa.

9 Logic implementation in CPLD

This section explains how to access CPLD registers for read/write operation using the IFC interface.

9.1 Writing to CPLD registers

This section explains how to perform a write operation on a CPLD register using IFC interface. The CPLD register write sequence starts with every positive edge of the IFC_WE_B signal and if the CPLD_CS signal is low, then data at the data bus is written to the addressed CPLD register as can be seen in the code snippet below.

```
always@(posedge ifc_we_b or negedge pwr_hrst_n or negedge
sw_rst_n)
    begin
        if (~pwr_hrst_n)                                  //Set CPLD registers to default value
            begin
                soft_mux_on[7:0] <= 8'b0;
                reg_rcw_src1[7:0] <= sw_rcw_src[7:0];          //SW4[1:8]
                reg_rcw_src2 <= sw_rcw_src[8];               //SW5[1]
                reg_bank[2:0] <= bank_sel[2:0];                //SW5[4:6]
                reg_tdmclk_mux_sel <= dipsw4_7;                //SW3[7]
                reg_sd1refclk_sel <= sd1refclk_sel;            //SW3[4]
                reg_uart_sel <= uart_sel;                    //SW3[3]
                reg_sysclk_sel <= sysclk_sel;                //SW5[2]
                global_rst <= 1'b0;
                reg_status_led <= 1'b0;
                reg_rtc <= 1'b0;
                reg_evdd_sel <= 1'b0;

                soft_mux_on1[7:0] <= 8'b0;
                reg_sdhc_vs_pinsel <= dipsw4_8;
            end
        else if (~sw_rst_n)
            begin
                system_rst <= 1'b0;
            end
        else if (~cpld_cs)
            begin
                case (cpld_addr[4:0])
                    3:          system_rst <= cpld_data[0];
                    4:          soft_mux_on[7:0] <= cpld_data[7:0];
                    5:          reg_rcw_src1[7:0] <= cpld_data[7:0];
                    6:          reg_rcw_src2 <= cpld_data[0];
                    7:          reg_bank[2:0] <= cpld_data[2:0];
                    8:          reg_sysclk_sel<= cpld_data[0];
                    9:          reg_uart_sel <= cpld_data[0];
                    10:         reg_sd1refclk_sel <= cpld_data[0];
                    11:         reg_tdmclk_mux_sel <= cpld_data[0];
                    13:         reg_status_led <= cpld_data[0];
                    14:         global_rst <= cpld_data[0];
                    16:         reg_rtc <= cpld_data[0];
                    17:         reg_evdd_sel <= cpld_data[0];
                    18:         soft_mux_on1[7:0] <= cpld_data[7:0];
                    19:         reg_sdhc_vs_pinsel <= cpld_data[0];
                    default:    non_reg <= cpld_data[0];
                endcase
            end
```

```
    end
endmodule
```

9.2 Timing diagram for CPLD write operation

The figure below shows the timing diagram for performing a write operation on a CPLD register using the IFC interface, when IFC is in Normal GPCM mode.

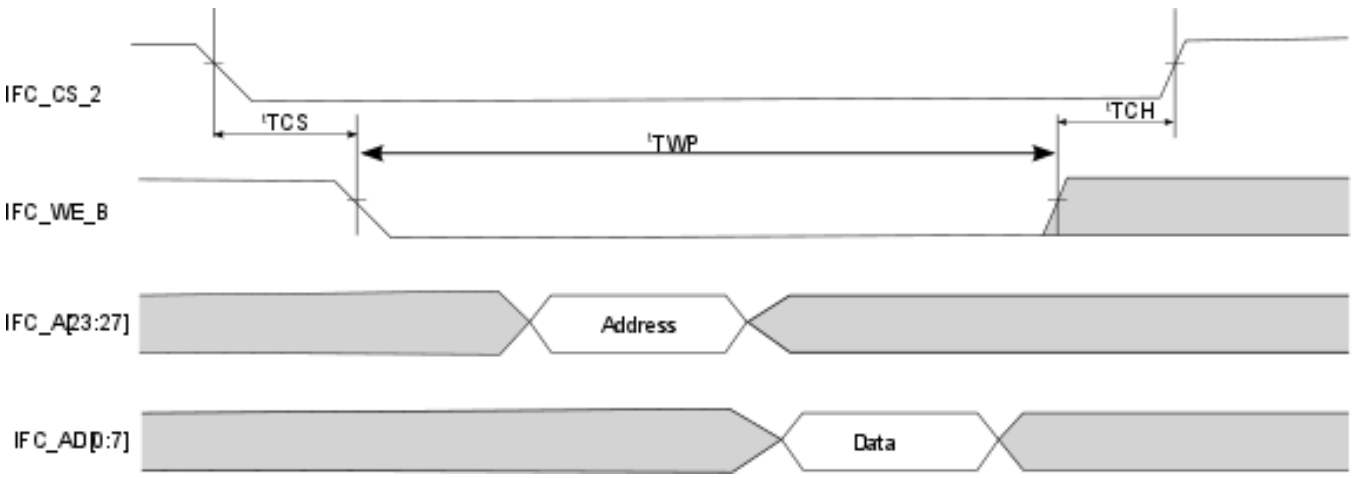

Figure 3. Normal GPCM: CPLD register write

The figure above shows the following two IFC write operation timing parameters:

- t_{TCS}: Chip-select assertion to WE assertion setup time
- t_{TWP}: Write enable pulse width
- t_{TCH}: Chip-select hold time with respect to WE deassertion

These parameters are programmable using Flash Timing register 2 for CSn - Normal GPCM Mode.

9.3 Reading from CPLD registers

This section explains how to perform a read operation on a CPLD register using IFC interface. The read cycle for a CPLD register is triggered when negative egde of the IFC_OE_B signal is detected as can be seen in the code snippet below.

```
always@(negedge ifc_oe_b)
    begin
        if (~cpld_cs)
            begin
                case (cpld_addr[4:0])
                    0:          regd[7:0] <= {4'b0, cpld_ver[3:0]};
                    1:          regd[7:0] <= {4'b0, cpld_ver_sub[3:0]};
                    2:          regd[7:0] <= {5'b0, pcb_ver[2:0] + 1};
                    3:          regd[7:0] <= {7'b0, system_rst};
                    4:          regd[7:0] <= soft_mux_on[7:0];
                    5:          regd[7:0] <= reg_rcw_src1[7:0];
                    6:          regd[7:0] <= {7'b0, reg_rcw_src2};
                    7:          regd[7:0] <= {5'b0, reg_bank[2:0]};
                    8:          regd[7:0] <= {7'b0, reg_sysclk_sel};
                    9:          regd[7:0] <= {7'b0, reg_uart_sel};
                    10:         regd[7:0] <= {7'b0, reg_sd1refclk_sel};
```

```
                11:         regd[7:0] <= {7'b0, reg_tdmclk_mux_sel};
                13:         regd[7:0] <= {7'b0, reg_status_led};
                14:         regd[7:0] <= {7'b0, global_rst};
                15:         regd[7:0] <= {7'b0, tdmr_prs_n};
                16:         regd[7:0] <= {7'b0, reg_rtc};
                17:         regd[7:0] <= {7'b0, reg_evdd_sel};
                18:         regd[7:0] <= soft_mux_on1[7:0];
                19:         regd[7:0] <= {7'b0, reg_sdhc_vs_pinsel};
                default:    regd[7:0]<=8'bzzzz_zzzz;
            endcase
        end
end
```

9.4 Timing diagram for CPLD read operation

The figure below shows the timing diagram for read operation on the CPLD register using the IFC interface, when IFC is in Normal GPCM mode.

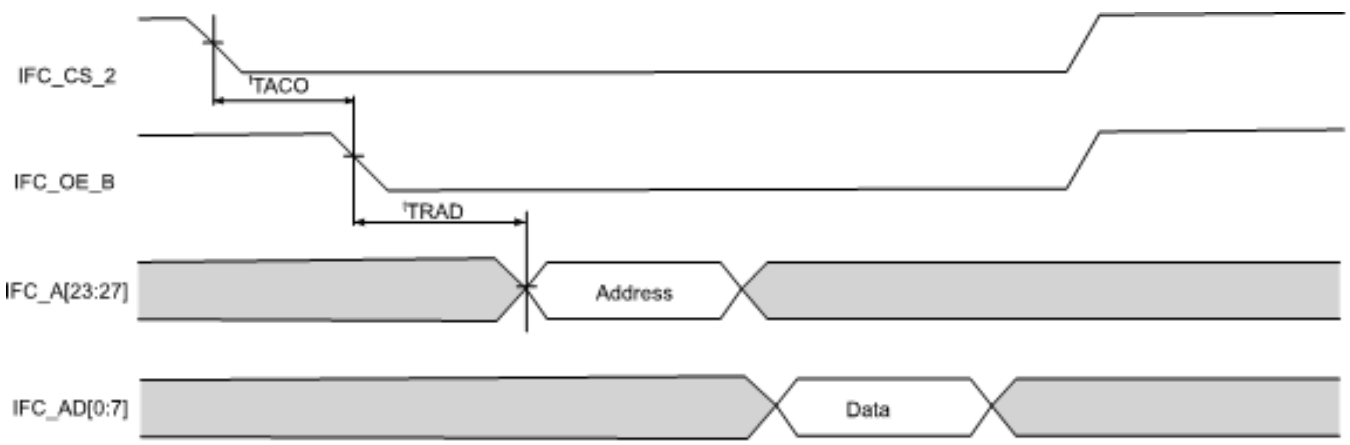

Figure 4. Normal GPCM: CPLD register read

The figure above shows the following two IFC read operation timing parameters:

- t_{TACO}: CS assertion to output enable (OE) assertion setup time
- t_{TRAD}: Output enable assertion time

These parameters are programmable using Flash Timing register 1 for CSn - Normal GPCM Mode.

10 Software implementation

NXP SDK 17.03 release supports communication between LS1043A processor and CPLD over IFC interface. This section shows code snippet from SDK 17.03 release demonstrating communication between the two devices.

Files of SDK1703 U-BOOT, which has code related to CPLD access, are listed below:

NOTE

SDK1703 U-BOOT path is: <SDK1703 installation directory>/build_ls1043ardb/tmp/work/ls1043ardb-fsl-linux/u-boot-qoriq/ 2016.01+fslgit-r0/git

- Header files:

- include/configs/ls1043ardb.h
- board/freescale/ls1043ardb/cpld.h

NOTE

The cpld.h file describes all the CPLD registers as a structure.

- C file:
  - board/freescale/ls1043ardb/cpld.c

Given below is the code snippet.

```
 u8 cpld_read(unsigned int reg)
{
        void *p = (void *)CONFIG_SYS_CPLD_BASE;

        return in_8(p + reg);
}

void cpld_write(unsigned int reg, u8 value)
{
        void *p = (void *)CONFIG_SYS_CPLD_BASE;

        out_8(p + reg, value);
}
```

How to Reach Us:

Home Page:
nxp.com

Web Support:
nxp.com/support

Document Number AN12098
Revision 0, 12/2017