Wolf Creek

Project

DDN

Customer

Wolf Creek Controller CPLD Firmware Specification

Product Name

JABIL	**Document Number:** DNPRO11301012469-60-06	**Revision: V1.0**
	Wolf Creek Controller CPLD Functional Specification	

Distribution List:	
Jabil Wolf Creek Project members	Jabil

Rev	Date	Originator(s)	Change Details
0.1	03/21/2014	Hudson Zhan	Initial revision
0.2	04/09/2014	Quake Wang	Remove ADR mode to support C2F feature
0.3	06/25/2014	Hudson Zhan	Update the document number; Update PIN output and register definition
0.4	08/20/2014	Hudson Zhan	Update all contents according to the schematic
0.5	08/25/2014	Hudson Zhan	Update the reset block and i2c registers and power sequencing
0.6	08/26/2014	Hudson Zhan	Add one UART from Psoc to Mux of UARTs. Add the section about shortcuts of switch serial console.
0.7	09/17/2014	Rich Wrenn (DDN), Hudson Zhan	1. Update CPLD Block diagram. 2. Change shortcut keys and add some setting register for switch UARTs to serial console. 3. Add POSTCODE debug register. 4. Update some grammar error and incorrect expression. 5. Add reset diagram.
0.8	09/22/2014	Rich Wrenn (DDN), Hudson Zhan	Update the section 7.3 about power load shedding.
0.9	12/03/2014	Rich Wrenn (DDN), Hudson Zhan	1. Update the hotkey value to switch UART port. 2. According to Green board schematic change, update section 7.3
1.0	12/06/2014	Rich Wrenn (DDN)	Default for register URT_INTERCONN_SET to 0x01 instead of ox00 in section 13.2.4

Approvals

Revision	Title	Name (Print)	Signature	Date
	PEL			
	Firmware Manager			
	Quality Assurance Engineer			
	Project Manager			
	Customer (if required)			
	(additional approvals- affected disciplines)			

Table of Contents

1. Purpose

This document specifies Wolf Creek Controller CPLD functionality requirements for Wolf Creek project. And it also leveraged the specification written by Intel for Grantley CPLD. The information contained in this document, along with information contained in referenced documents, is intended to provide all the information that is needed to design and develop the CPLD code. This document is intended to be used by vendors and internal developers.

2. Scope

This document is primarily intended for the development team responsible for designing and implementing the CPLD code, and for the test team responsible for validating the operation of the CPLD's operation. Additionally, this document will be useful to anyone needing to understand the function of the CPLD Firmware (e.g., hardware/software developers, team leaders, quality personnel, architects and customers).

3. Definitions/ Abbreviations

TERM	MEANING
ADR	**A**synchronous **D**RAM **R**efresh
ASCII	**A**merican **S**tandard **C**ode for **I**nformation **I**nterchange
BBU	**B**attery **B**ackup **U**nit
BMC	**B**oseboard **M**anagement **C**ontroller
C2F	**C**opy to **F**lash
CPLD	**C**omplex **P**rogrammable **L**ogic **D**evice
FPGA	**F**ield **P**rogrammable **G**ate Array
PSoC	**P**rogrammable Embedded **S**ystem-**O**n-**C**hip
I2C	**I**nter-**I**C **C**ommunication (Also known as TWI)
LPC	**L**ow **P**in **C**ount
NVSRAM	**n**on-**v**olatile **SRAM**
PSU	**P**ower **S**upply **U**nit
SATA	**S**erial **ATA**
SRAM	**S**tatic **R**andom **A**ccess **M**emory
UART	**U**niversal **A**synchronous **R**eceiver/**T**ransmitter
USB	**U**niversal **S**erial **B**us

4. Reference Documents

	Document Name	Version/Date	Author/Issuer
4.1	*Wolf Creek Product Requirements*	Revision 1.4 Mar 18, 2014	Jabil/Clay Wade
4.2	*Technical Note TN1204* *MachX02 Programming and Configuration Usage Guide*	February 2012	Lattice Se Clay Wade
4.3	*UM10204 - I2C-bus specification and user manual*	Rev. 03 Jun 19, 2007	NXP
4.4	*Intel Low Pin Count (LPC) Interface Specification*	Revision 1.1 August 2002	Intel
4.5	*Crystal Forest Platform Asynchronous DRAM Refresh (ADR)*	August 2010	Intel
4.6	Wolf Creek Power Behavior v01	January 30th	DDN

4.7	*Wolf Creek Product Requirements Document*	Revision: 2.0/ May 2, 2014	Jabil/Clay Wade

5. The Introduction of Main Feature

5.1 Introduction

Controller CPLD handles Grantley platform power on/down sequencing, reset timing control, NVSRAM support, mux of UARTs, LPC controller, POSTCODE decode, access to NVSRAM and I2C access for diagnostic and control.

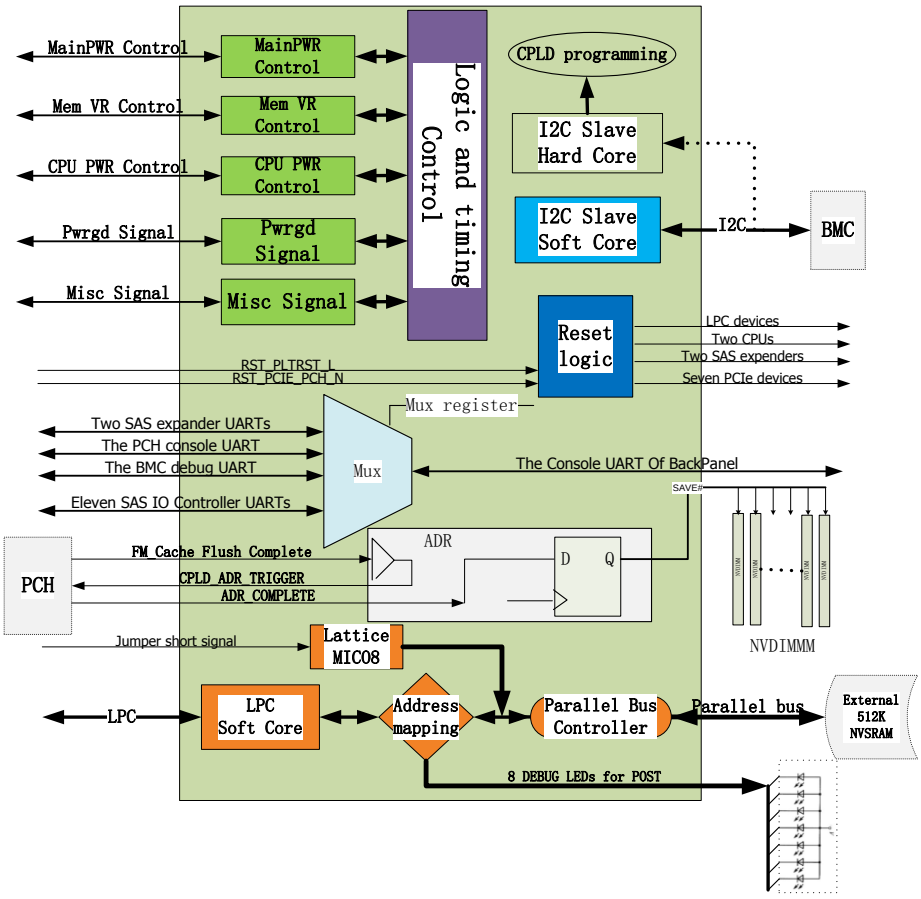


Figure 5.1 Wolf Creek Controller CPLD Block Diagram

5.2 Device Selection

A Lattice Semiconductor MachXO2 FPGA family device shall be used as the CPLD packaged in a 484 BGA.

6. Controller CPLD Pin-out Description and Pin Definition

6.1 Power sequencing Signal

	Direction	Signal Name of Schematic	Description
	Input	RST_RSMRST_N	Resume well reset signal
	Input	P1V26_STBY_PG	Standby power good signal
	Output	P1V05_STBY_EN	Standby power enable signal
	Input	P1V05_STBY_PG	Standby power good signal
	Input	PSU_ON_N	Main power good/enable signal
	Output	P12V_GD_CPLD_OUT	Main power good/enable signal
	Input	P12V_GD_R	Main power good/enable signal
	Output	PCIE_P12V_EN	Main power good/enable signal
	Output	P5V_HDD_EN	Main power good/enable signal
	Input	P5V_HDD_PG	Main power good/enable signal
	Output	P1V05_PCH_EN	Main power good/enable signal
	Input	P1V05_PCH_PG	Main power good/enable signal
	Output	PVCCIO_EN	Main power good/enable signal
	Input	PVCCIO_PG	Main power good/enable signal
	Output	P3V3_EN	Main power good/enable signal
	Input	P3V3_PG	Main power good/enable signal
	Output	PCIE_P3V3_EN	Main power good/enable signal
	Output	P1V5_PCH_EN	Main power good/enable signal
	Input	P1V5_PCH_PG	Main power good/enable signal
	Output	P1V8_LSI_EN	Main power good/enable signal
	Input	P1V8_LSI_PG	Main power good/enable signal
	Output	P1V8_PLX_EN	Main power good/enable signal
	Input	P1V8_PLX_PG	Main power good/enable signal
	Output	P2V5_PEX_EN	Main power good/enable signal
	Input	P2V5_PEX_PG	Main power good/enable signal
	Output	PVPP_ABCD_EN	Main power good/enable signal
	Input	PVPP_AB_PG	Main power good/enable signal
	Input	PVPP_CD_PG	Main power good/enable signal
	Output	PVPP_EFGH_EN	Main power good/enable signal
	Input	PVPP_EF_PG	Main power good/enable signal
	Input	PVPP_GH_PG	Main power good/enable signal
	Output	P1V5_LSI_EN	Main power good/enable signal
	Input	P1V5_LSI_PG	Main power good/enable signal
	Output	P1V0_EN	Main power good/enable signal
	Input	P1V0_PG	Main power good/enable signal
	Output	PVDDQ_ABCD_EN	Main power good/enable signal
	Input	PVDDQ_AB_PG	Main power good/enable signal
	Input	PVDDQ_CD_PG	Main power good/enable signal
	Output	PVDDQ_EFGH_EN	Main power good/enable signal
	Input	PVDDQ_EF_PG	Main power good/enable signal
	Input	PVDDQ_GH_PG	Main power good/enable signal
	Output	8056_P0V92_EN	Main power good/enable signal
	Input	8056_P0V92_0_PG	Main power good/enable signal
	Input	8056_P0V92_1_PG	Main power good/enable signal
	Output	PLX9749_VDD09_EN	Main power good/enable signal
	Input	PLX9749_VDD09_PG	Main power good/enable signal

	Direction	Signal Name of Schematic	Description
	Output	PVTT_ABCD_EN	Main power good/enable signal
	Input	PVTT_AB_PG	Main power good/enable signal
	Input	PVTT_CD_PG	Main power good/enable signal
	Output	PVTT_EFGH_EN	Main power good/enable signal
	Input	PVTT_EF_PG	Main power good/enable signal
	Input	PVTT_GH_PG	Main power good/enable signal
	Output	PVCCIN_CPU0_EN	Main power good/enable signal
	Input	PVCCIN_CPU0_PG	Main power good/enable signal
	Output	PVCCIN_CPU1_EN	Main power good/enable signal
	Input	PVCCIN_CPU1_PG	Main power good/enable signal
	Output	PLX_P0V9_0_EN	Main power good/enable signal
	Output	PLX_P0V9_1_EN	Main power good/enable signal
	Output	PLX_P0V9_2_EN	Main power good/enable signal
	Output	PLX_P0V9_3_EN	Main power good/enable signal
	Output	PLX_P0V9_4_EN	Main power good/enable signal
	Output	PLX_P0V9_5_EN	Main power good/enable signal
	Input	PLX_P0V9_0_PG	Main power good/enable signal
	Input	PLX_P0V9_1_PG	Main power good/enable signal
	Input	PLX_P0V9_2_PG	Main power good/enable signal
	Input	PLX_P0V9_3_PG	Main power good/enable signal
	Input	PLX_P0V9_4_PG	Main power good/enable signal
	Input	PLX_P0V9_5_PG	Main power good/enable signal
	Output	LSI_P0V9_0_EN	Main power good/enable signal
	Output	LSI_P0V9_1_EN	Main power good/enable signal
	Output	LSI_P0V9_2_EN	Main power good/enable signal
	Output	LSI_P0V9_3_EN	Main power good/enable signal
	Output	LSI_P0V9_4_EN	Main power good/enable signal
	Output	LSI_P0V9_5_EN	Main power good/enable signal
	Output	LSI_P0V9_6_EN	Main power good/enable signal
	Output	LSI_P0V9_7_EN	Main power good/enable signal
	Output	LSI_P0V9_8_EN	Main power good/enable signal
	Output	LSI_P0V9_9_EN	Main power good/enable signal
	Input	LSI_P0V9_0_PG	Main power good/enable signal
	Input	LSI_P0V9_1_PG	Main power good/enable signal
	Input	LSI_P0V9_2_PG	Main power good/enable signal
	Input	LSI_P0V9_3_PG	Main power good/enable signal
	Input	LSI_P0V9_4_PG	Main power good/enable signal
	Input	LSI_P0V9_5_PG	Main power good/enable signal
	Input	LSI_P0V9_6_PG	Main power good/enable signal
	Input	LSI_P0V9_7_PG	Main power good/enable signal
	Input	LSI_P0V9_8_PG	Main power good/enable signal
	Input	LSI_P0V9_9_PG	Main power good/enable signal
	Input	LSI_P0V9_10_PG	Main power good/enable signal
	Output	USB3_PWRON1	Main power good/enable signal
	Output	USB3_PWRON2	Main power good/enable signal
	Output	PWRGD_CPU0_LVC3	CPU power good signal
	Output	PWRGD_CPU1_LVC3	CPU power good signal
	Input	PWRGD_CPUPWRGD	CPU power good signal
	Output	PWRGD_DRAMPWRGD_CO	Dram power good signal

	Direction	Signal Name of Schematic	Description
	Output	PWRGD_SYS_PWROK	System power good signal
	Output	CPLD_PWRGD_PCH_APWROK	PCH power good signal
	Output	CPLD_PWRGD_PCH_PWROK	PCH power good signal
	Input	XDP_CPU_SYSPWROK	XDP signal
	Input	XDP_PWRGD_RST_N	XDP signal
	Input	XDP_RST_CO_N	XDP signal
	Output	LAN1_VR_EN	LAN VR power enable signal

6.2 RESET Signals

6.3	Direction	Signal Name of Schematic	Description
	Input	RST_PLTRST_N	Platform reset signal
	Output	CPLD_BMC_PCIE_RESET_N	PCIe reset signal from CPLD
	Output	CPLD_BMC_RESET_R_N	LPC reset signal
	Output	ETH_SWITCH_RESET_N	Ethernet reset signal
	Output	TPM_RESET_BUF_N	TPM reset signal
	Output	RST_CPU0_LVC3_N	CPU0 reset signal
	Output	RST_CPU1_LVC3_N	CPU1 reset signal
	Output	RST_DLY_CPURST_LVC3	CPU reset signal
	Input	RST_PCIE_CPU_N	PCIe reset signal from CPU
	Output	RST_PCIE_PCH_LAN1_N	PCIe reset signal to LAN
	Input	RST_PCIE_PCH_N	PCIe reset signal from PCH
	Output	PRI_SEP_RST_N	SAS 8005 reset signal
	Output	SEC_SEP_RST_N	SAS 8005 reset signal
	Output	PCIE_SLOT1_RST_N	PCIE slot reset signal
	Output	PCIE_SLOT2_RST_N	PCIE slot reset signal
	Output	PCIE_SLOT3_RST_N	PCIE slot reset signal
	Output	PE_8608_PERST_BUF_N	PEX8608 reset signal
	Output	PE_RST_DRV2_L	PCIe drive reset signal
	Output	PE_RST_DRV4_L	PCIe drive reset signal
	Output	LSI0_SYS_RST_N0	LSI3008 reset signal
	Output	LSI1_SYS_RST_N0	LSI3008 reset signal
	Output	LSI2_SYS_RST_N0	LSI3008 reset signal
	Output	LSI3_SYS_RST_N0	LSI3008 reset signal
	Output	LSI4_SYS_RST_N0	LSI3008 reset signal
	Output	LSI5_SYS_RST_N0	LSI3008 reset signal
	Output	LSI6_SYS_RST_N0	LSI3008 reset signal
	Output	LSI7_SYS_RST_N0	LSI3008 reset signal
	Output	LSI8_SYS_RST_N0	LSI3008 reset signal
	Output	LSI9_SYS_RST_N0	LSI3008 reset signal
	Output	LSI10_SYS_RST_N0	LSI3008 reset signal

6.4 Presence signals and MISC

	Direction	Signal Name of Schematic	Description
	Input	FM_SLPA_N	Reserved for future
	Input	FM_SLPS0_N	Reserved for future
	Input	FM_SLPS3_N	Reserved for future
	Input	FM_SLPS4_N	Reserved for future
	Input	FM_SLPS5_N	Reserved for future

	Input	FM_CPU1_SKTOCC_LVT3_N	SKTOCC signal of CPU1
	Input	FM_CPU1_THERMTRIP_LVC3_N	Thermal trip signals of CPU1
	Input	FM_CPU_ERR1_CO_N	Error signal of CPU
	Output	FM_ERR1_DLY_N	Error delay signal of CPU
	Output	FM_LVC3_THERMTRIP_DLY	Thermal trip delay signals
	Input	FM_PCH_PRSNT_CO_N	Presence signal from PCH
	Input	FM_THERMTRIP_CO_N	Presence signal from PCH
	Input	FP_RST_BTN_N	Reserved for future
	Input	PWR_BTN_LED	Illuminate the LED to indicate the button is pressed
	Input	PE_8608_FATAL_ERR_BUF_R_N	PEX8608 fatal error signal
	Input	PE_8608_PEX_INTA_BUF_N	PEX8608 interrupt signal

6.5 ADR Signals

	Direction	Signal Name of Schematic	Description
	Input	AC_FAIL_R_N	ADR logic signals
	Input	AC_GOOD_R_LATCHED_N	ADR logic signals
	Input	ADR_COMPLETE_R	ADR logic signals
	Input	ADR_STATUS	ADR logic signals
	Output	CPLD_ADR_TRIGGER_R	ADR logic signals
	Input	FM_CACHE_FLUSH_COMPLETE	ADR logic signals
	input	FM_SMI_ACTIVE_N	ADR logic signals
	input	FM_SMI_INT_CPLD_N	ADR logic signals
	input	MCU_R_INIT	ADR logic signals
	input	FM_BACKUP_MODE_PCH	ADR logic signals
	input	PBG_CPLD_R_CNTL0	ADR logic signals
	input	PBG_CPLD_R_CNTL1	ADR logic signals
	Output	R_SAVE_CHA_DIMM0	Trigger signal to NVDIMM enter SAVE state
	Output	R_SAVE_CHA_DIMM1	Trigger signal to NVDIMM enter SAVE state
	Output	R_SAVE_CHB_DIMM0	Trigger signal to NVDIMM enter SAVE state
	Output	R_SAVE_CHB_DIMM1	Trigger signal to NVDIMM enter SAVE state
	Output	R_SAVE_CHC_DIMM0	Trigger signal to NVDIMM enter SAVE state
	Output	R_SAVE_CHC_DIMM1	Trigger signal to NVDIMM enter SAVE state
	Output	R_SAVE_CHD_DIMM0	Trigger signal to NVDIMM enter SAVE state
	Output	R_SAVE_CHD_DIMM1	Trigger signal to NVDIMM enter SAVE state
	Output	R_SAVE_CHF_DIMM0	Trigger signal to NVDIMM enter SAVE state
	Output	R_SAVE_CHF_DIMM1	Trigger signal to NVDIMM enter SAVE state
	Output	R_SAVE_CHG_DIMM0	Trigger signal to NVDIMM enter SAVE state
	Output	R_SAVE_CHG_DIMM1	Trigger signal to NVDIMM enter SAVE state
	Output	R_SAVE_CHH_DIMM0	Trigger signal to NVDIMM enter SAVE state
	Output	R_SAVE_CHH_DIMM1	Trigger signal to NVDIMM enter SAVE state

6.6 UARTS Mux Signals

6.7	Direction	Signal Name of Schematic	Description
	Input	MUX_UART_RX	RX signal of serial console bus for debug
	Output	MUX_UART_TX	TX signal of serial console bus for debug
	Input	PRI_SEP_UART_RX	RX signal of primary SAS expender UART bus
	Output	PRI_SEP_UART_TX	TX signal of primary SAS expender UART bus

	Input	BMC_UART1_RX	RX signal of CPU Complex UART bus
	Output	BMC_UART1_TX	TX signal of CPU Complex UART bus
	Input	BMC_UART_RXD5	RX signal of BMC debug UART bus
	Output	BMC_UART_TXD5	TX signal of BMC debug UART bus
	Input	SEC_SEP_UART_RX	RX signal of secondary SAS expender UART bus
	Output	SEC_SEP_UART_TX	TX signal of secondary SAS expender UART bus
	Output	HS_SWD_SWV	Reserved signal
	Output	PWR_CPLD_PB37D_PSOC	TX signal of PSoC debug UART bus
	Input	IOC1_UART_BUF_RX	RX signal of SAS3008 #1 UART bus
	Output	IOC1_UART_BUF_TX	TX signal of SAS3008 #1 UART bus
	Input	IOC2_UART_BUF_RX	RX signal of SAS3008 #2 UART bus
	Output	IOC2_UART_BUF_TX	TX signal of SAS3008 #2 UART bus
	Input	IOC3_UART_BUF_RX	RX signal of SAS3008 #3 UART bus
	Output	IOC3_UART_BUF_TX	TX signal of SAS3008 #3 UART bus
	Input	IOC4_UART_BUF_RX	RX signal of SAS3008 #4 UART bus
	Output	IOC4_UART_BUF_TX	TX signal of SAS3008 #4 UART bus
	Input	IOC5_UART_BUF_RX	RX signal of SAS3008 #5 UART bus
	Output	IOC5_UART_BUF_TX	TX signal of SAS3008 #5 UART bus
	Input	IOC6_UART_BUF_RX	RX signal of SAS3008 #6 UART bus
	Output	IOC6_UART_BUF_TX	TX signal of SAS3008 #6 UART bus
	Input	IOC7_UART_BUF_RX	TX signal of SAS3008 #7 UART bus
	Output	IOC7_UART_BUF_TX	RX signal of SAS3008 #7 UART bus
	Input	IOC8_UART_BUF_RX	TX signal of SAS3008 #8 UART bus
	Output	IOC8_UART_BUF_TX	RX signal of SAS3008 #8 UART bus
	Input	IOC9_UART_BUF_RX	TX signal of SAS3008 #9 UART bus
	Output	IOC9_UART_BUF_TX	RX signal of SAS3008 #9 UART bus
	Input	IOC10_UART_BUF_RX	TX signal of SAS3008 #10 UART bus
	Output	IOC10_UART_BUF_TX	RX signal of SAS3008 #10 UART bus
	Input	IOC11_UART_BUF_RX	TX signal of SAS3008 #11 UART bus
	Output	IOC11_UART_BUF_TX	RX signal of SAS3008 #11 UART bus

6.8 LPC Signals

	Direction	Signal Name of Schematic	Description
	Input	CLK_33M_CPLD_LPC	Clock signal of LPC bus
	Input	LPC_FRAME_N	Frame signal of LPC bus
	Input/output	LPC_LAD<0>	Data signal of LPC bus
	Input/output	LPC_LAD<1>	Data signal of LPC bus
	Input/output	LPC_LAD<2>	Data signal of LPC bus
	Input/output	LPC_LAD<3>	Data signal of LPC bus

6.9 NVSRAM Signals

	Direction	Signal Name of Schematic	Description
	Output	NVSRAM_A<0:18>	19 address signals to NVSRAM
	Input/output	NVSRAM_D<0:7>	8 data signals to NVSRAM
	Output	NVSRAM_CE_N	Chip enable signal to NVSRAM
	Input/output	NVSRAM_HSB_N	Hardware STORE busy signal of NVSRAM
	Output	NVSRAM_OE_N	Output enable signal to NVSRAM
	Output	NVSRAM_WE_N	Write enable signal to NVSRAM

	Input	NVSRAM_CLR_N	One Jumper signal to erase NVSRAM

6.10 I2C Signals

	Direction	Signal Name of Schematic	Description
AA9	Input	BMC_SMB_8_R_SCL	SCL for Enclosure Management I2C Bus used to access the debug register data.
Y10	Input/output	BMC_SMB_8_R_SDA	SDA for Enclosure Management I2C Bus used to access the debug register data.
A13	Input	BMC_SMB_8_SCL	SCL for Programming I2C bus
B13	Input/output	BMC_SMB_8_SDA	SDA for Programming I2C bus

6.11 CPLD global Signals and debug Signals

	Input	PWR_CPLD_CLK_25M	25MHz clock input
	Input	CPLD_RESET_N	POR signals from reset chip
	Output	SLOT_ID0	Slot id signal to indicate the current slot is slot 1 or slot 2
	Output	LED_POSTCODE_0	LED signal for BIOS POST code and debug
	Output	LED_POSTCODE_1	LED signal for BIOS POST code and debug
	Output	LED_POSTCODE_2	LED signal for BIOS POST code and debug
	Output	LED_POSTCODE_3	LED signal for BIOS POST code and debug
	Output	LED_POSTCODE_4	LED signal for BIOS POST code and debug
	Output	LED_POSTCODE_5	LED signal for BIOS POST code and debug
	Output	LED_POSTCODE_6	LED signal for BIOS POST code and debug
	Output	LED_POSTCODE_7	LED signal for BIOS POST code and debug

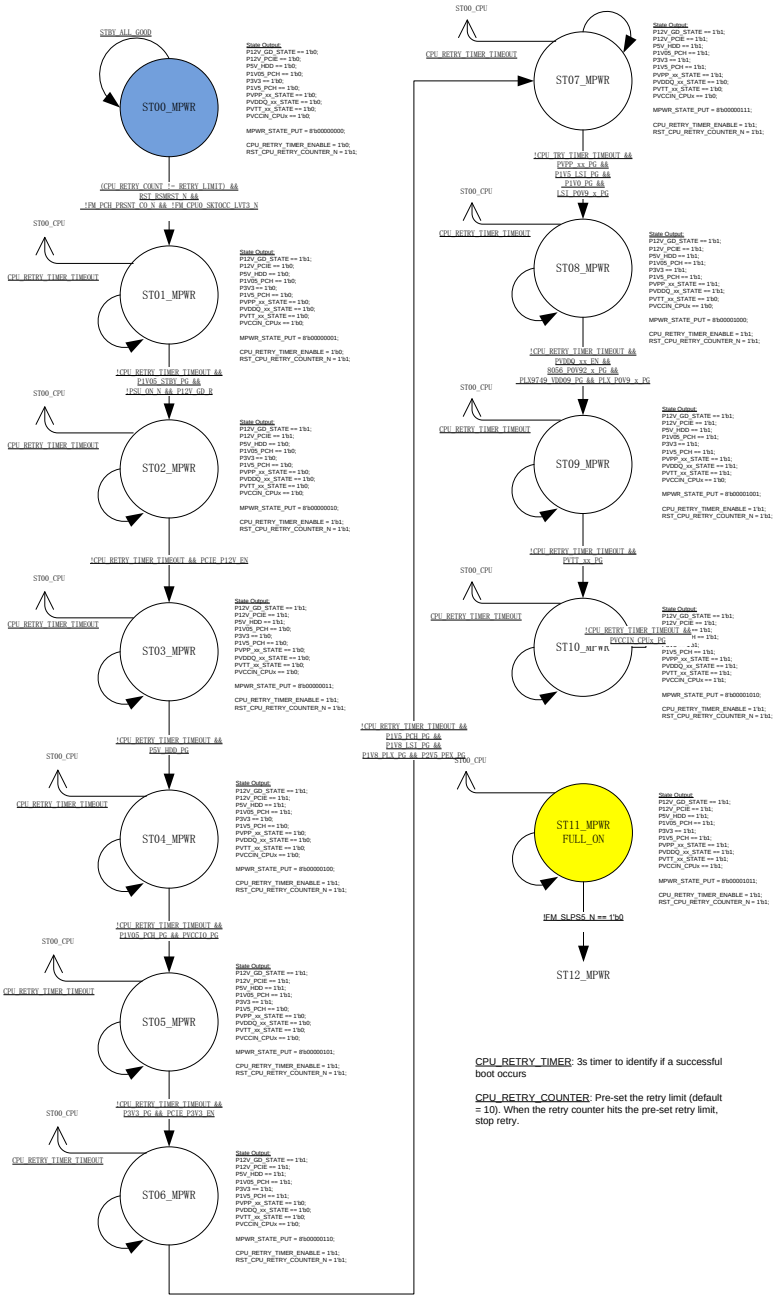
6.12 JTAG Interface Signals

	Direction	Identifier	Description
	Input	JTAG_CPLD_EN_L	Lattice JTAGENB signal
	Input/output	JTAG_CPLD_INIT_L	Lattice INITN signal
	Input	JTAG_CPLD_PROGRAMN	Lattice PROGRAMN signal
	Input	JTAG_CPLD_TCK	TCK signal for the IEEE1149.1 (JTAG) interface
	Input	JTAG_CPLD_TMS	TMS signal for the IEEE1149.1 (JTAG) interface
	Input	PWR_CPLD_TDI	TDI signal for the IEEE1149.1 (JTAG) interface
	Output	PWR_MI_CPLD_R_TDO	TDO signal for the IEEE1149.1 (JTAG) interface

7. Power Sequencing Block

7.1 CPU Complex Power-ON Sequencing State Machine

In Wolf Creek controller system, there includes the standby power rails and main power rails. In the standby power rails timing, hardware circuits have sequenced every standby power rails. CPLD shall handle main power rails timing. The power-on state machine of the system main power rails always keeps the initial step ST00_MPWR unless any one of standby power rails is stable. The state machine transition shall move under the condition of the trigger events. Figure 7.1 shows Power-ON Sequencing State Machine State Transition.



CPU_RETRY_TIMER: 3s timer to identify if a successful boot occurs

CPU_RETRY_COUNTER: Pre-set the retry limit (default = 10). When the retry counter hits the pre-set retry limit, stop retry.

Figure 7.1 Power-ON Sequencing State Machine State Transition

7.2 CPU Complex Power-Down Sequencing State Machine

The power-down sequencing state machine basically is the inverse of the power-on sequencing state machine. But it shall more simply shut down some power rails.

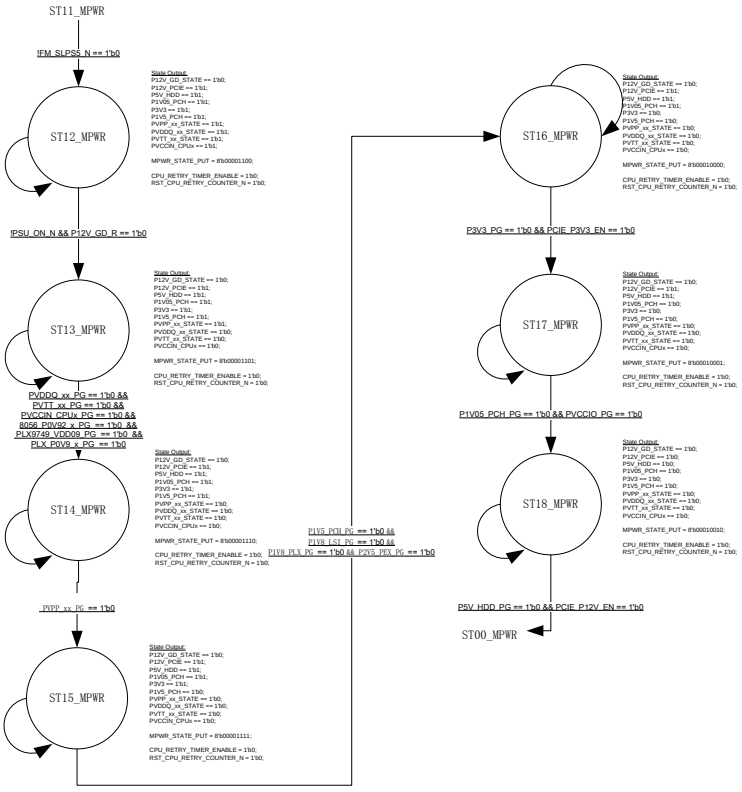


Figure 7.2 Power-Down Sequencing State Machine State Transition

7.3 Power Load Shedding

CPLD supplies one register (PWR_LOAD_SHED, 0x30h) for the BMC (or SFA OS via the BMC) to shut down the devices by group. The devices are grouped into four colors, purple, brown, blue, green, which are shown in the block diagram below.

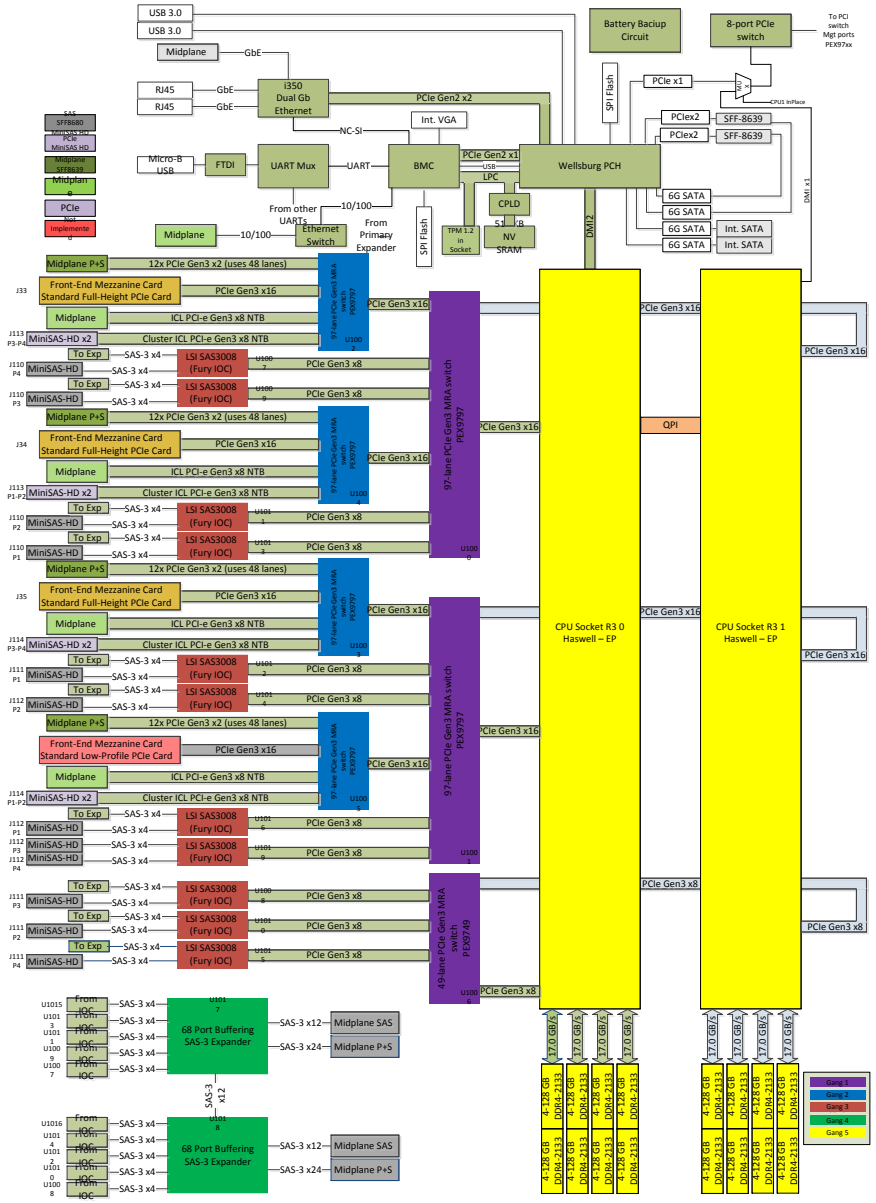


Figure 7.3 System Block Diagram

The register PWR_LOAD_SHED shall enable/disable the VRMs of the four groups. To assert the enable signals PLX_P0V9_0_EN and PLX9749_VDD09_EN of purple devices, bit0 of the register PWR_LOAD_SHED shall be set to 1b by SFA OS instruct BMC to send i2c write command. Others colors devices are similar to purple devices. The register is defined in section 13.2.4.8.

Device Color	Parts	VRMs Enable Signals	VRMs Power Good
Purple	U1000: PLX9797	PLX_P0V9_0_EN	PLX_P0V9_0_PG
	U1001: PLX9797	PLX_P0V9_1_EN	PLX_P0V9_1_PG
	U1006: PLX9749	PLX9749_VDD09_EN	PLX9749_VDD09_PG
Blue	U1002: PLX9797	PLX_P0V9_2_EN	PLX_P0V9_2_PG
	U1003: PLX9797	PLX_P0V9_3_EN	PLX_P0V9_3_PG
	U1004: PLX9797	PLX_P0V9_4_EN	PLX_P0V9_4_PG
	U1005: PLX9797	PLX_P0V9_5_EN	PLX_P0V9_5_PG
brown	U1007: LSI SAS3008	LSI_P0V9_0_EN	LSI_P0V9_0_PG
	U1008: LSI SAS3008	LSI_P0V9_1_EN	LSI_P0V9_1_PG
	U1009: LSI SAS3008	LSI_P0V9_2_EN	LSI_P0V9_2_PG
	U1010: LSI SAS3008	LSI_P0V9_3_EN	LSI_P0V9_3_PG
	U1011: LSI SAS3008	LSI_P0V9_4_EN	LSI_P0V9_4_PG
	U1012: LSI SAS3008	LSI_P0V9_5_EN	LSI_P0V9_5_PG
	U1013: LSI SAS3008	LSI_P0V9_6_EN	LSI_P0V9_6_PG
	U1014: LSI SAS3008	LSI_P0V9_7_EN	LSI_P0V9_7_PG
	U1015: LSI SAS3008	LSI_P0V9_8_EN	LSI_P0V9_8_PG
	U1016: LSI SAS3008	LSI_P0V9_9_EN	LSI_P0V9_9_PG
	U1019: LSI SAS3008	LSI_P0V9_10_EN	LSI_P0V9_10_PG
Green	Primary PM8056	8056_P0V92_EN	8056_P0V92_0_PG
	Secondary PM8056		8056_P0V92_1_PG

Table 7.3 the Enable and Power good signals of the Four Groups of devices.

After fire hose dump (FHD) is completed and data is stored, BMC shall force all devices to shut down and the system to be shut down by disabling P12V hot plug, regardless of the devices and OS left alive. CPLD still shall execute the Power Down sequence of section 7.2, to turn down the system and devices.

7.4 The timing of Power Good signals between Wellsburg and Haswell

The CPLD shall be responsible for driving the Power Good signals for the CPU system.

7.4.1 The timing of PCH_APWROK

The CPLD shall drive the signal CPLD_PWRGD_PCH_APWROK to Wellsburg at Minimum 1ms after the signal P1V05_STBY_PG is good.

7.4.2 The timing of PCH_PWROK

The CPLD shall drive the signal CPLD_PWRGD_PCH_PWROK to Wellsburg no less than100ms after the signal P1V5_PCH_PG is good. Also, receive the signal DRAM_PWROK from Wellsburg under all the right power rails of memory.

7.4.3 The timing of SYS_PWROK

On coming of PVCCIN_CPU0_PG/ PVCCIN_CPU1_PG from the PWM power controller, and

PWRGD_CPUPWRGD from Haswell CPU, CPLD shall drive the signal PWRGD_SYS_PWROK Wellsburg chip in the time of 1ms.

7.4.4 The timing of CPU PWRGD Signals

Upon receiving CPLD_PWRGD_PCH_PWROK, Wellsburg will drive the PWRGD_CPUPWRGD signal. The CPLD, in turn, shall drive PWRGD_CPU0_LVC3 and PWRGD_CPU1_LVC3 to CPU0, CPU1.

PWRGD_CPU0_LVC3 = ~FM_CPU0_SKTOCC_LVT3_N and PWRGD_CPUPWRGD

PWRGD_CPU1_LVC3 = ~FM_CPU1_SKTOCC_LVT3_N and PWRGD_CPUPWRGD

7.5 The timing of Cater Error and Thermal Trip

7.5.1 FM_CPU_ERR1_CO_N Delay function

Input: FM_CPU_ERR1_CO_N
Output: FM_ERR1_DLY_N
Require the FM_CPU_ERR1_CO_N signal to be buffered through 400us (200us - 500us) delay circuit before being connected to the corresponding Wellsburg GPIO signal.

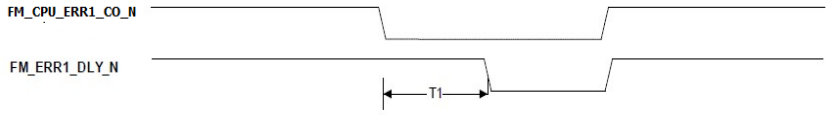


Figure 6 Timing Diagram for FM_ERR1_N Delay
Note: T1 = 200 uSec to 500 uSec
In order to minimize the logic utilization, CPLD implementation will allow the time delay to vary between 200 uS to 400 uS.

7.5.2 FM_THERMTRIP_CO_N Delay function

Input: FM_ THERMTRIP_CO_N
Output: FM_LVC3_THERMTRIP_DLY
Require the FM_THERMTRIP_CO_N signal to be buffered through 400us (200us - 500us) delay circuit before being connected to the corresponding Wellsburg GPIO signal.
In order to minimize the logic utilization, CPLD implementation will allow the time delay to vary between 200 uS to 400 uS.

8. System reset Block

8.1 Reset Diagram and Reset signals

8.1.1 Reset overview

For every device in the Wolf Creek system, all the devices shall be kept reset until their individual power rails are normal stable state. So, CPLD shall start to release the reset signals in the start of RST_PLTRST_N released from Wellsburg.

8.1.2 Reset Diagram

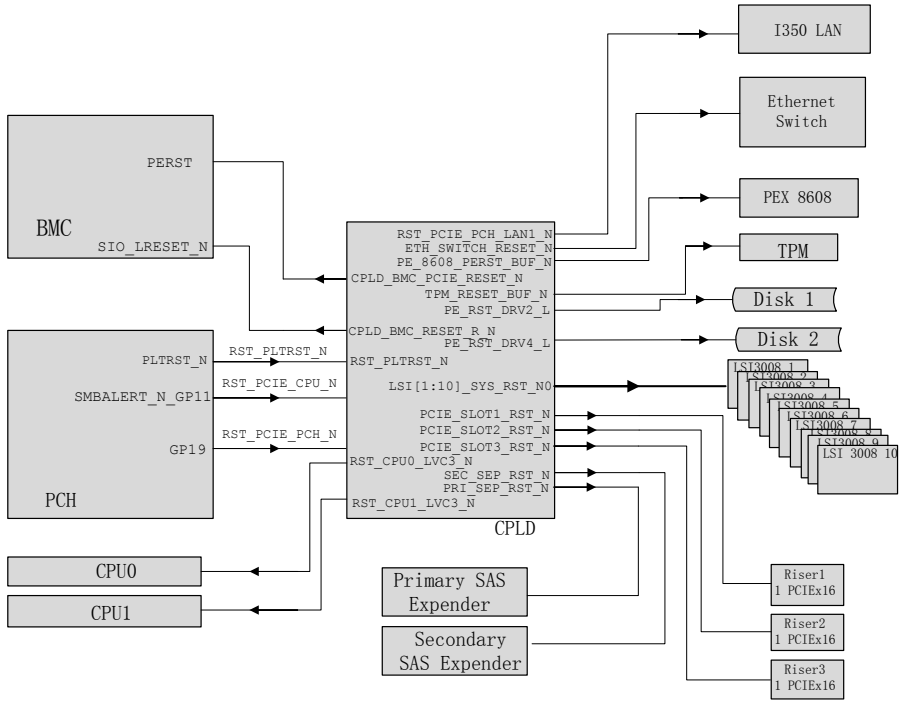


Figure 8.1 System Reset Diagram

8.2 CPU Devices Reset Function

When receive the event of RST_PLTRST_N released, CPLD shall release the reset signals of two CPUs.

RST_CPU0_LVC3_N = RST_PLTRST_L and RST_XDP_L after 5ms.
RST_CPU1_LVC3_N = RST_PLTRST_L and RST_XDP_L after 5ms.

Then, CPLD shall release the reset of LPC devices and

CPLD_BMC_RESET_R_N = RST_PLTRST_L after 5ms.
TPM_RESET_BUF_N = RST_PLTRST_L after 5ms.

Base on release the reset of RST_PLTRST_L, after the release of RST_PCIE_CPU_N or RST_PCIE_PCH_N from Wellsburg, CPLD shall the all the reset of PCIe devices.

ETH_SWITCH_RESET_N = RST_PLTRST_L AND RST_PCIE_PCH_N after 1ms.
CPLD_BMC_PCIE_RESET_N = RST_PLTRST_L AND RST_PCIE_PCH_N after 1ms.
RST_PCIE_PCH_LAN1_N = RST_PLTRST_L AND RST_PCIE_PCH_N after 1ms.
ETH_SWITCH_RESET_N = RST_PLTRST_L AND RST_PCIE_PCH_N after 1ms.

PCIE_SLOT1/2/3_RST_N = RST_PLTRST_L AND RST_PCIE_CPU_N.
PE_8608_PERST_BUF_N = RST_PLTRST_L AND RST_PCIE_CPU_N.
LSI[0:10]_SYS_RST_N0 = RST_PLTRST_L AND RST_PCIE_CPU_N.
PE_RST_DRV2/4_L = RST_PLTRST_L AND RST_PCIE_CPU_N.

Also, CPLD shall reset other devices after RST_PLTRST_N.

LPC devices: CPLD_BMC_RESET_R_N = RST_PLTRST_N.

TPM_RESET_BUF_N = RST_PLTRST_L.

8.3 SAS expender resets

For waiting for CPU system power on, CPLD shall release the reset of primary and secondary SAS expender 5ms after the signal RST_PLTRST_L released.

PRI_SEP_RST_N = RST_PLTRST_L after 5ms.

SEC_SEP_RST_N = RST_PLTRST_L after 5ms.

9. NVDIMM Feature Support

9.1 NVDIMM Power Fail Process

The Wolf Creek platform must preserve the contents of a large cache in the case of a power failure. When the system uses NVDIMMs to preserve the data, CPLD also is involved in the procession of data preservation. Logic flow shall go through the following steps:

1. AC power fails and AC_OK (or AC_FAIL_N) is de-asserted by the PSOC. AC_OK = ! ((PSU_A_AC_OK_L & PSU_B_AC_OK_L) | BBU_ON_BATTERY).
2. AC_OK (or AC_FAIL_N) de-assertion triggers the NMI of PCH GPIO7.
3. The DDN OS completes the data protection, and then asserts PCH GPIO72 to notify Controller CPLD to assert ADR_TRIGGER connected to PCH GPIO37.
4. ADR_TRIGGER will flush all CPU memory controller buffers and places the DIMMs into self-refresh mode and then asserts ADR_COMPLETE via PCH specific pin RESERVED5.
5. ADR_COMPLETE is connected to Controller CPLD to drive the DIMM FORCE_SAVE# pin of all DIMM sockets. At this point, the DIMM sockets are powered via Vmem from the Ultra capacitor Battery Pack Module and FORCE_SAVE# is latched within the DIMMs.
6. The DIMMs copy their contents to on-board flash using power from the Ultra capacitors. When done, power to the DIMMs is shut off.

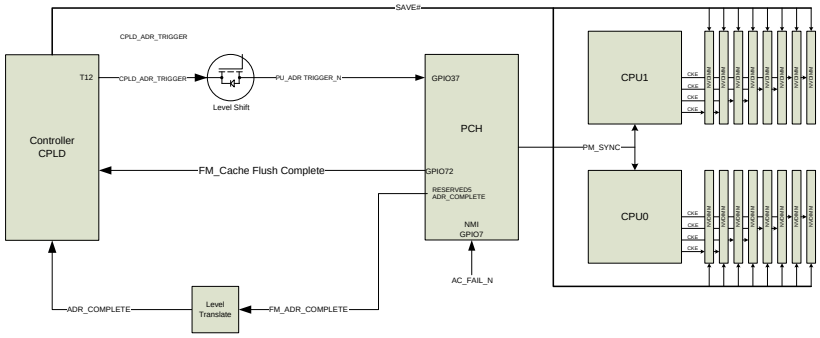


Figure 9.1 NVDIMM Power Fail Process

9.2 CPLD effort

When the OS asserts FM_CACHE_FLUSH_COMPLETE, the CPLD shall assert CPLD_ADR_TRIGGER to trigger ADR actions by GPIO 37 on PCH.

When ADR_COMPLETE is asserted, the CPLD shall right now drive SAVE# pin to trigger the DIMM to copy data to flash.

10. Multiplex and Switch of Serial Interfaces

10.1 MUX Control registers

The Wolf Creek Controller CPLD has a UART Multiplexor (MUX) that can be used to create a connection between the microUSB connector on the backplate of the Controller canister and FTDI USB-to-UART converter to one of 16 UARTs within the canister. They respectively are connected to the eleven SAS IO Controller UART interfaces, the two SAS expander UART interfaces, the CPU Complex UART interface, the BMC debug UART interface, and the PSoC debug UART interface.

This MUX is controlled by the following registers

1. A nonvolatile register, URT_INTERCONN_SET (0xA4h), whose value can be changed by an IPMI command to the BMC which the BMC passes via I2C to the Power CPLD. URT_INTERCONN_SET defines the default UART setting. During manufacturing, URT_INTERCONN_SET is set to the value that selects the BMC UART (i.e., the main processor console).
2. A nonvolatile register, URT_KEY_DISABLE_SET (0xA5h), whose value can be changed by an IPMI command to the BMC which the BMC passes via I2C to the Power CPLD. URT_KEY_DISABLE_SET determines whether the URT_KEY value should be ignored. During manufacturing URT_KEY_DISABLE_SET is set to ENABLED.
3. A volatile register, URT_KEY (0xA6h), whose value can be set by typing a key sequence on the terminal (or terminal emulator) connected to the microUSB connector. The Power-On-Reset value of URT_KEY is DEFAULT.
4. A volatile register, URT_KEY_LOCK (0xA7h), whose value can be change by an I2C command from the Primary SAS Expander to the BMC which the BMC passes via I2C to the Power CPLD. URT_KEY_LOCK determines whether hot keys should modify URT_KEY. The Power-On-Reset value of URT_KEY_LOCK is UNLOCKED.

10.2 Logic function of UARTs Switch

In the following cases, UARTs switch shall be determined by different registers or hot keys.

1. If URT_KEY_DISABLE_SET is DISABLED then URT_INTERCONN_SET determines the MUX setting.
2. If URT_KEY_DISABLE_SET is ENABLED and UART_KEY is DEFAULT then URT_INTERCONN_SET determines the MUX setting.
3. If URT_KEY_DISABLE_SET is ENABLED and URT_KEY has a value other than DEFAULT then URT_KEY determines the MUX setting.
4. If URT_KEY_LOCK is UNLOCKED and URT_KEY_DISABLE_SET is ENABLED then URT_KEY can be set with a 2-character control sequence starting with a CTRL-^ (ASCII RS) (CTRL-SHIFT-6 on a US keyboard) according to the table below.
    a. If a valid second character (per the table below) is typed within 3 seconds of the CTRL-^ then the Power CPLD will modify URT_KEY and will not pass either character to the selected UART.
    b. If no character is typed within 3 seconds of the CTRL-^ or if an invalid character is typed after the CTRL-^ then the CTRL-^ and any invalid character typed within the 3 second window will be passed to the selected UART and URT_KEY will not be modified.
    c. If CTRL-^ is typed twice within 3 seconds then a single CTRL-^ will be passed to the UART and URT_KEY will not be modified.
5. If URT_KEY_LOCK is LOCKED or URT_KEY_DISABLE_SET is DISABLED then URT_KEY cannot be modified by hot keys and all characters are passed to the UART.

Table 10.3: The Hot Key of Switch UARTs to serial console

UART Selected by MUX	Hot Key Sequence	ASCII Code Sequence	URT_KEY Register Value

DEFAULT (Select UART specified in URT_INTERCONN_SET)	<CTRL-^> + z	0x1E + 0x7A	0xFF
BMC debug UART	<CTRL-^> + a	0x1E + 0x61	0
CPU Complex UART	<CTRL-^> + b	0x1E + 0x62	1
Primary SAS expender UART	<CTRL-^> + c	0x1E + 0x63	2
Secondary SAS expender UART	<CTRL-^> + d	0x1E + 0x64	3
PSoC debug UART	<CTRL-^> + e	0x1E + 0x65	4
SAS3008 #1 UART	<CTRL-^> + f	0x1E + 0x66	5
SAS3008 #2 UART	<CTRL-^> + g	0x1E + 0x67	6
SAS3008 #3 UART	<CTRL-^> + h	0x1E + 0x68	7
SAS3008 #4 UART	<CTRL-^> + i	0x1E + 0x69	8
SAS3008 #5 UART	<CTRL-^> + j	0x1E + 0x6A	9
SAS3008 #6 UART	<CTRL-^> + k	0x1E + 0x6B	10
SAS3008 #7 UART	<CTRL-^> + l	0x1E + 0x6C	11
SAS3008 #8 UART	<CTRL-^> + m	0x1E + 0x6D	12
SAS3008 #9 UART	<CTRL-^> + n	0x1E + 0x6E	13
SAS3008 #10 UART	<CTRL-^> + o	0x1E + 0x6F	14
SAS3008 #11 UART	<CTRL-^> + p	0x1E + 0x70	15

10.3 Serial Console Setting

The following parameters are required as configuration to access the serial console:

Formatted: Space After: 0.5 line

- 115,200bps band rate
- 8 bits per character
- 1 stop bit
- No parity
- None Flow Control

10.4 The Instantiation of EFB

Within the on-chip flash memory, there is 256K bits of non-volatile storage, the User Flash Memory Block (UFM), which is used for a variety of applications. In Wolf Creek, controller CPLD shall save and restore the UARTs Connection Setting Register (URT_INTERCONN_SET, 0xA5h) and UARTs Shortcut Key Disable Setting Register (URT_KEY_DISABLE_SET, 0xA6h) to the flash memory.

To activate and access the User Flash Memory Block (UFM), CPLD needs instantiate one interface Embedded Function Block (EFB) to enable WISHBONE slave interface to access UFM memory. CPLD shall implement one WISHBONE master interface to interact with the EFB WISHBONE slave interface. The WISHBONE master interface continues to save the two registers of UARTs into the second and third bytes of UFM memory, but the first byte is fixed to 0x28, which is only used to identify the memory location.

Every time the CPLD powers up or a POR happens, CPLD, through the WISHBONE master interface, shall retore the second and third bytes into URT_INTERCONN_SET, 0xA5h and URT_KEY_DISABLE_SET, 0xA6h. But UARTs connection status always is determined by logic of UARTs MUX in CPLD.

Below shows the EFB interface of CPLD chip from Lattice MachXO2 datasheet.

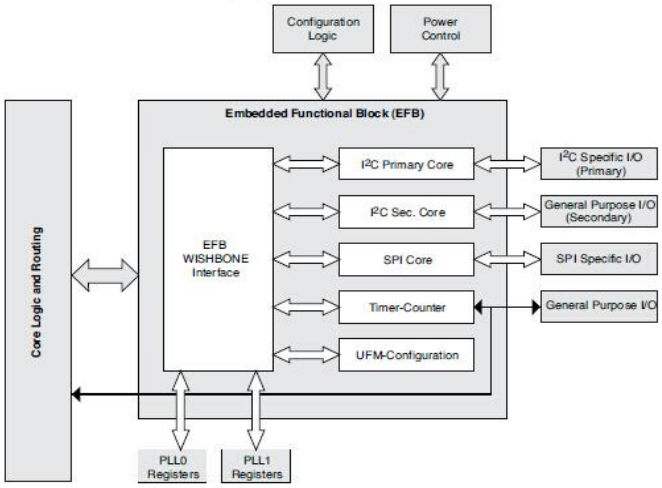


Figure 10-2 EFB interface diagram of Lattice MachXO2

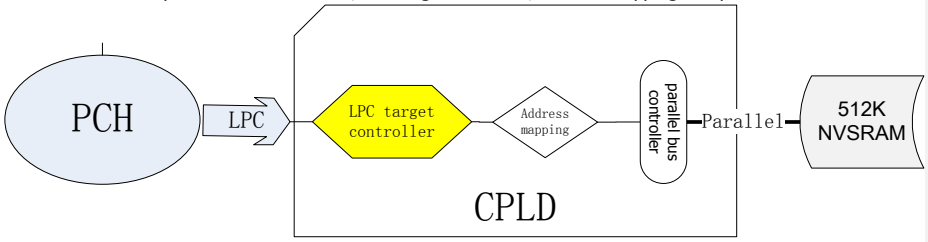
11. NVSRAM Access Support through LPC

11.1 Overview

The Wolf Creek Controller CPLD supports I/O Read/Write Cycles and Memory Read/Write cycles on LPC bus. I/O read/write cycles are used to program the base address of the NVSRAM memory map on the LPC bus. Memory read/write cycles are used to access the memory mapped address by LPC.

The Controller CPLD parallel interface provides access to external Cypress 512k Bytes NVSRAM. The NVSRAM has a 25ns – 45ns access time, which allows the software to read and write individual bytes with a latency of less than 1 microsecond from LPC bus.

CPLD shall implement three modules, LPC target controller, address mapping and parallel bus controller.



11.2 LPC I/O Transactions

I/O Cycles must be used to program the decode address for the CPLD LPC Memory Read/Write register map. The CPLD will not decode memory read/write cycles until all 32 bits of the decode address have been programmed.

Table 10-1: LPC I/O Ports

I/O Port Address	Register	Access Mode	Default	Description
900h	LPC BAR[7:0]	RO	00h	LPC base address registers [23:16].
901h	LPC BAR[15:8]	RW	F8h	LPC Base address registers [31:24]

11.3 LPC Memory Read/Write Transactions

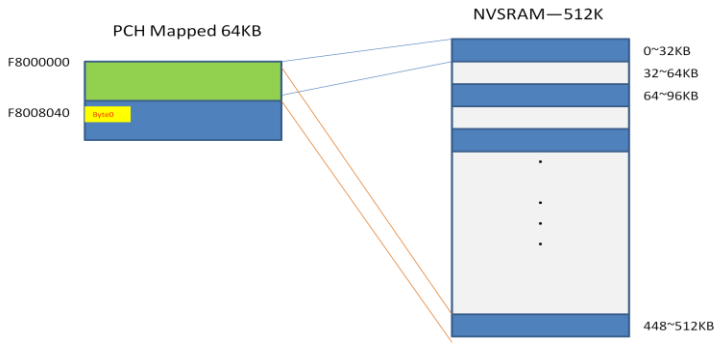
The Controller CPLD decodes LPC memory read/write operations targeting addresses that fall within the configured range for the LPC memory map.

11.4 Parallel Interface

According to the interface of NVSRAM and requirement, the parallel interface includes 8 bits of data, 19 bits of address, chip selection, write enable, output enable, and hardware STORE busy. Only 8-bit accesses are supported.

11.5 Address Mapping

In Wellsburg system, PCH assigns one 64k bytes memory space to LPC bus from 0xF8000000 ~0xF800FFFF. CPLD shall cut out one half space 32k bytes to map NVSRAM space, which address space is defined into the start address 0xF8000000 with the end of 0xF8007FFF. In the spare address space, one byte page register is defined in the address 0xF008040. The content of the page register is one value of 0-15. Below the chart shows the PCH mapped 64k bytes to NVSRAM 512k bytes.



The 512k bytes NVSRAM is equally divided into 16 address segments. Each segment has 32k bytes space as one page. So, as the value of the page register, the page of 32k bytes in NVSRAM shall be mapped into the system.

Table 10.1: The mapping of system memory space and NVSRAM as page register

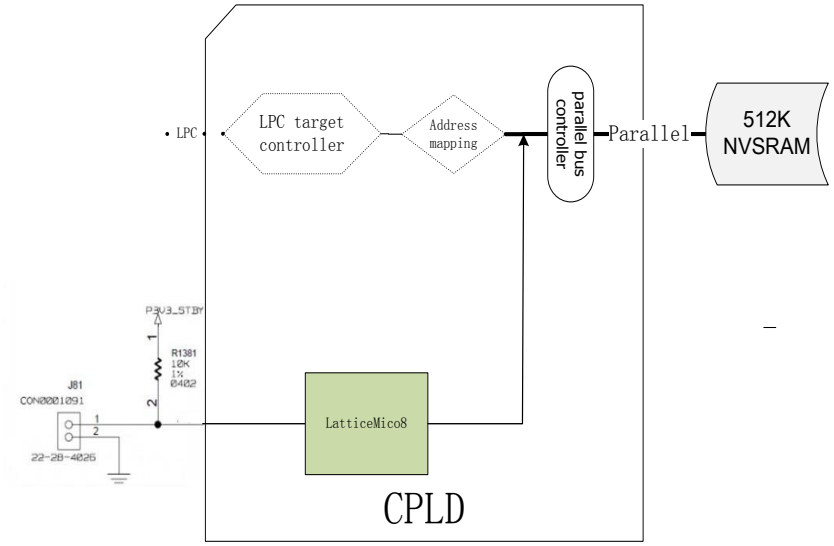
The memory space of System for LPC	The Value of Page Register (0xF8008040)	The segment of NVSRAM
0xF8000000～0xF8007FFF	0	0x00000～0x07FFF
0xF8000000～0xF8007FFF	1	0x08000～0x0FFFF
0xF8000000～0xF8007FFF	2	0x10000～0x17FFF
0xF8000000～0xF8007FFF	3	0x18000～0x1FFFF
0xF8000000～0xF8007FFF	4	0x20000～0x27FFF
0xF8000000～0xF8007FFF	5	0x28000～0x2FFFF
0xF8000000～0xF8007FFF	6	0x30000～0x37FFF
0xF8000000～0xF8007FFF	7	0x38000～0x3FFFF
0xF8000000～0xF8007FFF	8	0x40000～0x47FFF
0xF8000000～0xF8007FFF	9	0x48000～0x4FFFF

0xF8000000～0xF800FFFF	10	0x50000～0x57FFF
0xF8000000～0xF8007FFF	11	0x58000～0x5FFFF
0xF8000000～0xF8007FFF	12	0x60000～0x67FFF
0xF8000000～0xF8007FFF	13	0x68000～0x6FFFF
0xF8000000～0xF8007FFF	14	0x70000～0x77FFF
0xF8000000～0xF8007FFF	15	0x78000～0x7FFFF
0xF8008000～0xF800FFFF	x	None

So, when software accesses to NVSRAM, it firstly need write into the page register to indicating some segment of NVSRAM, then read data from or write data to this segment of NVSRAM memory space.

12. The Behavior of Jumper

There is one jumper (J81) designed to clear the external NVSRAM chip in Wolf Creek system. In the case that user adds one cap on the jumper after plug out the controller canister from the chassis of the Wolf Creek system, CPLD shall clear once the entire NVSRAM chip every time the CPLD complete power on reset (POR).



The 8-bit microcontroller LatticeMico8 is built up in the cost of less than 1000 LUTs resource. One C code in MICO8 for the feature of writing zeros into all the address 0x00000-0x7FFFF of NVSRAM, shall run once after CPLD powers up. But, if disconnects the short to J81, LatticeMICO8 shall not do any action for NVSRAM.

For software to use, one register of 0xA0 is defined to identify whether the jumper have been added or not.

13. I2C bus Descriptions

13.1 I2C bus overall

The I2C port on the CPLD shall provide an I2C-bus interface that is compliant with the I2C-bus specification and user manual [REF 3] and operate under the following parameters:

- Only I2C Slave bus operations shall be supported.
- The I2C Slave address shall be:
  - I2C Ports 1 C8/C9h

- I2C programming Port: 80/81h

- The bus speed shall be Standard-mode (Sm), with a bit rate of 100kbits/s [REF 3 section 5].
- Only 7-bit addressing shall be used. 10-bit addressing [REF 3 section 3.1.11] shall not be supported.
- Sequential read/write shall not be supported.
- General call address [REF 3section 3.1.13] shall not be supported.
- Software reset [REF 3 section 3.1.14] shall not be supported.

13.2 Signal State Access and Control

The state of selected input and output signals that are connected to the CPLD can be only read, and some can be controlled by accessing registers in the CPLD. Each CPLD register contains fields that are associated with a CPLD input or output signal. The current state of a signal is obtained by reading the register that contains that signals data. In the case of output signals, the state of a signal can be controlled by writing to the register that contains that signals data.

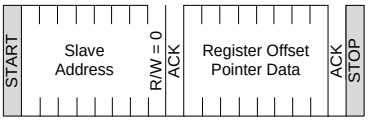
The CPLD registers are accessed using an I2C interface to the CPLD. The following sections describe how signal states can be read and/or modified using the CPLD register set.

13.2.1 Register Offset Pointer

The CPLD shall maintain a Register Offset Pointer which is used to indicate which CPLD register is to be accessed. This value shall represent an offset into the CPLD register map. The value of the Register Offset Pointer shall be the register offset of the last register accessed incremented by one.

The default value of the Register Offset Pointer following a reset of the CPLD due to power-on or assertion of the CPLD_RESET input signal shall be 00h.

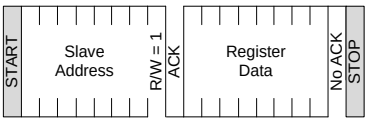
The value of the Register Offset Pointer can be set using an I2C write operation. The first byte received from an I2C Master following the slave address-R/W byte during a write operation (i.e., when the R/W bit is set to 0b) shall replace the current Register Offset Pointer value. This operation is illustrated as follows:



13.2.2 Register Read Operations

13.2.2.1 Current Register Read

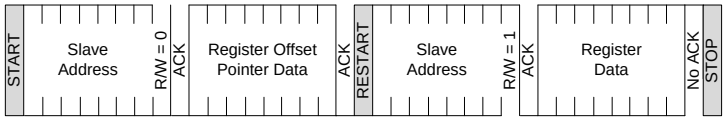
To read the data from the register currently pointed to by the Register Offset Pointer the I2C Master sends a START then Slave address-R/W byte with the R/W bit is set to 1b. The CPLD will acknowledge this and send the data for the register pointed to by the Register Offset Pointer. The Master will then terminate the transfer with a STOP condition as show below without acknowledging the byte.



If the current Register Offset Pointer value points at an unused CPLD register offset then FFh shall be returned as the register data.

13.2.2.2 Random Register Read

To read the data from the register that is not currently pointed to by the Register Offset Pointer the I2C Master first performs a dummy write operation to set the Register Offset Pointer to the offset of the register to be read. The I2C Master sends a START then Slave address-R/W byte with the R/W bit is set to 0b, which the CPLD will acknowledge. The I2C Master then sends a byte containing the new value for the Register Offset Pointer, which again the CPLD acknowledges. Following this a read operation is preformed using a RE-START. The I2C Master sends a START then Slave address-R/W byte with the R/W bit is set to 1b. The CPLD will acknowledge this and send the data for the register pointed to by the new Register Offset Pointer that was just received. The Master will then terminate the transfer with a STOP condition without acknowledging the byte.



If the new Register Offset Pointer value points at an unused CPLD register offset then FFh shall be returned as the register data.

13.2.3 Register Write Operations

13.2.3.1 Register Write

A register write operation will by default include setting the value of the Register Offset Pointer. The I2C Master sends a START then Slave address-R/W byte with the R/W bit is set to 0b, which the CPLD will acknowledge. The I2C Master then sends a byte containing the new value for the Register Offset Pointer, which again the CPLD acknowledges. Following this the I2C Master sends a byte that contains the data to be written into the register currently pointed to by the Register Offset Pointer. After the CPLD writes the received data into the register it issues an ACK and the I2C Master then terminates the transfer with a STOP condition.

If the new Register Offset Pointer value points at an invalid CPLD register offset then no data shall be written to any CPLD register and the Register Offset Pointer shall not be incremented.

13.2.4 Register Map

13.2.4.1 Device ID Register (DEVICE_ID, 0x00h-0x01h)

Offset	Register	Field	Description	Type	Reset (Default)
01h-00h	DEVICE_ID	bit<15:0>	The DDN Device ID is a binary value that is assigned on a product by product basis so that the type of device can be identified by clients of the functions contained in the CPLD.	Read/write	0x2590

13.2.4.2 Chip Number ID Register (CHIP_ID, 0x02h)

Offset	Register	Field	Description	Type	Reset (Default)

02h	CHIP_ID	bit<7:0>	Chip Number ID Register Identifies the chip number of the controller CPLD 1 = Controller CPLD; 2 = Mid plane interface CPLD; 3 = IOM CPLD; 4 = Reserved; 5 = Baseboard CPLD 1; 6 = Baseboard CPLD 2; 7 = System Status Module CPLD; *others=reserved*.	Read Only	0x01

13.2.4.3 CPLD Major Version Number Register (CPLD_MAJ_VER, 0x03h)

Offset	Register	Field	Description	Type	Reset (Default)
03h	CPLD_MAJ_VER	bit<7:0>	CPLD Major Version Number Register Identifies the major revision of the Controller CPLD	Read Only	x

13.2.4.4 CPLD Minor Version Number Register (CPLD_MIN_VER, 0x04h)

Offset	Register	Field	Description	Type	Reset (Default)
04h	CPLD_MIN_VER	bit<7:0>	CPLD Minor Version Number Register Identifies the minor revision of the Controller CPLD	Read Only	x

13.2.4.5 CPLD Header Information Checksum Register (CPLD_HDR_CHK, 0x05h)

Offset	Register	Field	Description	Type	Reset (Default)
05h	CPLD_HDR_CHK	bit<7:0>	CPLD Header Information Checksum Register One binary value that contains a value so that the 8-bit summation of register offset 00h through register offset 05h equals 00h.	Read Only	x

13.2.4.6 CPU Complex Power on sequencing State Machine State Transition Registers (CPU_PWRON_STAT, 0x20h-0x2Fh)

Offset	Register	Field	Description	Type	Reset (Default)
20h-2Fh	CPU_PWRON_STAT	bit<7:0>	CPU Complex Power on sequencing State Machine State Transition Registers Representing the current State of CPU main power rails.	Read Only	0x00h

There are sixteen 8-bit registers implemented to record the last 16 state transition of the CPU Complex Power-On Sequencing State Machine. Each 8-bit register follows the format below.

To read offset 0x2F, the current state of the CPU Complex Power-On Sequencing State Machine can be retrieved. To read offset 0x2E, the previous state transaction can be known…understand from which state the current state was moved. See more in the section 7.1

13.2.4.7 CPU Complex Power Down sequencing State Machine State Transition Registers (CPU_PWRDWN_STAT, 0x30h-0x3Fh)

Offset	Register	Field	Description	Type	Reset (Default)
30h-3Fh	CPU_PWRDWN_STAT	bit<7:0>	CPU Complex Power-Down sequencing State Machine State Transition Registers Representing the current State of CPU main power rails.	Read Only	0x00h

There are sixteen 8-bit registers implemented to record the last 16 state transition of the CPU Complex Power-Down sequencing State Machine. Each 8-bit register follows the format above.

Reading offset 0x3F reports the current state of the CPU Complex Power-Down Sequencing State Machine. Reading offset 0x3F reports the previous state of the CPU Complex Power-Down Sequencing State Machine, and thus can be used to understand from which state the current state was moved. See more in the section 7.2

13.2.4.8 Power Load Shedding Registers (PWR_LOAD_SHED, 0x30h)

Offset	Register	Field	Description	Type	Reset (Default)
30h	PWR_LOAD_SHED	bit<7:0>	Power Load Shedding Registers: The register functions to turn on/off the purple/blue/brown/green devices for fire hose dump, through enable/disable the VRMs of the devices. All the devices default to enable VRMs. More is detailed in section 7.3 Bit<0>: the bit is used to enable/disable VRMs of the purple devices. The default value is 0b to disable. Bit<1>: the bit is used to enable/disable VRMs of the blue devices. The default value is 0b to disable. Bit<2>: the bit is used to enable/disable VRMs of the brown devices. The default value is 0b to disable. Bit<3>: the bit is used to enable/disable VRMs of the green devices. The default value is 0b to disable. Bit<7-4>: Reserved	Read and Write	0x0Fh

Note, when the system power on/off, in the power sequence, the power load shedding registers shall be ANDed into turn on/off the power of the devices. Only the corresponding bit is ON to turn on the devices during power up.

13.2.4.9 NVSRAM Jumper State Registers (NVSRAM_JPR_STAT, 0xA0h)

Offset	Register	Field	Description	Type	Reset (Default)

A0h	NVSRAM_JPR_STAT	bit<7:0>	NVSRAM Jumper State Registers 0: Not add the jumper, the signal NVSRAM_CLR_N keeps the high state. 1: Added the jumper, the signal NVSRAM_CLR_N is asserted into the low state.	Read Only	0x00h

13.2.4.10 UARTs Connecting Status Register (URT_CONNECTING, 0xA4h)

Offset	Register	Field	Description	Type	Reset (Default)
A4h	URT_CONNECTING	bit<7:0>	UARTs Connecting Status Registers: It indicates the current UART is being connected to which one UART console. BMC UART is default to connect to UART console. 0x00 : BMC UART 0x01 : CPU Complex UART 0x02 : Primary SAS expender UART 0x03 : Secondary SAS expender UART 0x04 : PSoC debug UART 0x05 : SAS3008 #1 UART 0x06 : SAS3008 #2 UART 0x07 : SAS3008 #3 UART 0x08 : SAS3008 #4 UART 0x09 : SAS3008 #5 UART 0x0A : SAS3008 #6 UART 0x0B : SAS3008 #7 UART 0x0C : SAS3008 #8 UART 0x0D : SAS3008 #9 UART 0x0E : SAS3008 #10 UART 0x0F : SAS3008 #11 UART others : reserved	Read Only	0x01h

13.2.4.11 UARTs Connection Setting Register (URT_INTERCONN_SET, 0xA5h)

Offset	Register	Field	Description	Type	During Manufacturing

A5h	URT_INTERCONN_SET	bit<7:0>	UART Connection Setting Register, a nonvolatile register, could be set by an IPMI command to the BMC which the BMC passes to the CPLD. 0x00 : BMC UART 0x01 : CPU Complex UART 0x02 : Primary SAS expender UART 0x03 : Secondary SAS expender UART 0x04 : PSoC debug UART 0x05 : SAS3008 #1 UART 0x06 : SAS3008 #2 UART 0x07 : SAS3008 #3 UART 0x08 : SAS3008 #4 UART 0x09 : SAS3008 #5 UART 0x0A : SAS3008 #6 UART 0x0B : SAS3008 #7 UART 0x0C : SAS3008 #8 UART 0x0D : SAS3008 #9 UART 0x0E : SAS3008 #10 UART 0x0F : SAS3008 #11 UART Others : reserved, these values shall be prohibited from being written to the register.	Read and write	0x01h

13.2.4.12 UARTs Shortcut Key Disable Setting Register (URT_KEY_DISABLE_SET, 0xA6h)

Offset	Register	Field	Description	Type	During Manufacturing
A6h	URT_KEY_DISABLE_SET	bit<7:0>	URT_KEY_DISABLE_SET Register, a nonvolatile register, could be set by an IPMI command to the BMC which the BMC passes to the CPLD. 0: Enable use shortcut key to switch UARTs to the serial console 1: Disable use shortcut key to switch UARTs to the serial console. Others: reserved, and it also is prohibited to be written	Read and write	0x01h

13.2.4.13 UARTs Shortcut Keys Value Register (URT_KEY, 0xA7h)

Offset	Register	Field	Description	Type	Reset (Default)

A7h	URT_KEY	bit<7:0>	UARTs Shortcut Keys Value Register, a volatile and only readable register, whose value can be set by typing a key sequence on the terminal connected to the microUSB connector. Valid value : 0～15(0x00-0x0F); Default value : 256 (0xFF);	Read Only	0xFFh

13.2.4.14 UARTs Shortcut Keys Lock Register (URT_KEY_LOCK, 0xA8h)

Offset	Register	Field	Description	Type	Reset (Default)
A8h	URT_KEY_LOCK	bit<7:0>	UARTs Shortcut Keys Lock Register, a volatile register, whose value could be set by an IPMI command to the BMC which the BMC passes to the CPLD. 0: Unlock shortcut key to switch UARTs to the serial console 1: Lock shortcut key to switch UARTs to the serial console. Others: reserved, and it also is prohibited to be written	Read and Write	0x00h

13.2.4.15 POST CODE DEBUG Register (POSTCODE_DBG, 0xADh)

Offset	Register	Field	Description	Type	Reset (Default)
ADh	POSTCODE_DBG	bit<7:0>	POST CODE LED Registers: It reflects the late POST code decoded to I/O port 0x80.	Read Only	0x00h

While the system is booting, the BIOS outputs a series of POST codes to I/O Port 0x80. In the process of LPC transmission, CPLD shall decode I/O write to I/O port 0x80. The late POST code shall be stored into the register POSTCODE_DBG, and display the hexadecimal value via 8 debug LEDs.

14. Programmability

This section describes the device used for the Controller CPLD design and the available programming paths.

14.1 CPLD Programming

The CPLD shall support the following Lattice MachX02 programming and configuration modes (refer to *MachX02 Programming and Configuration Usage Guide* [REF 1]):

- JTAG (using the CPLD ispJTAG port)
- I2C (using the CPLD I2C Update Port)

14.2 ISP (in-system programming) online mode with PC

Before starting to program the CPLD, the user must connect to the CPLD JTAG connector on the baseboard to a PC with ispDOWNLOAD Cable HW-USBN-2A. Power on main board, and enter standby mode or working mode for supply power to CPLD device.

User get ready for compiled JED format image, and programs JED file into HDDS CPLD device via Lattice ispVM system. For more information, refer to the document "UG48 ispDOWNLOAD Cable user's Guide" about ispVM system in lattice website.
The standard JTAG header pin-out and description is shown in the table below.

ispDOWNLOAD Cable Pin Definitions	**Name**	**Pin Type**	**Description**
VCC	Programming Voltage	Input	Connect to VCC plane of the target device. Typical ICC = 10mA. The board design supplies the power for VCC.
TDO	Test Data Output	Input	Used to shift data out via the IEEE1149.1 (JTAG) programming standard.
TDI	Test Data Input	Output	Used to shift data in via the IEEE1149.1 programming standard.
ispEN/PROG	Enable	Output	Enable device to be programmed.
TMS	Test Mode Select Input	Output	Used to control the IEEE1149.1 state machine.
GND	Ground	Input	Connect to ground plane of the target device
TCK	Test Clock Input	Output	Used to clock the IEEE1149.1 state machine
INIT	Initialize	Input	Indicates that ORCA device is ready for configuration.
DONE	Done	Input	An open-collector signal that indicates when configuration is complete.

The JTAG port is also wired to the BMC connector and may be programmed from the BMC.

14.3 ISP (in-system programming) online mode with BMC

BMC on the controller canister can upgrade the Wolf Creek Controller CPLD using the I2C bus. The BMC will support this function only when BMC is alive. Refer to CPLD upgrade portion of the BMC specification. To activate the fresh image downloaded into CPLD internal flash, BMC shall send the REFRESH command by using I2C, after complete to download the entire data into CPLD.

14.3.1 CPLD Internal Register for Upgrade

Registers	I2C action	Description
0xF2	Read only	CPLD online program Module version code, Default: 01h
0xF8	Write only	Package transfer, Current package size: 1 byte
0xF9	Read only	Status register, Bit [7-1]: Reserved Bit [0]: FIFO status bit 0 – FIFO not empty, need to wait 50ms, 5 seconds for time out. 1 – FIFO is empty, continue upgrade
0xFA	Write only	Action Register, Bit [7-1]: Reserved Bit [0]: 1 - Start Programming

Table 21 Internal Register for Upgrade