

牛芯半导体（深圳）有限公司

**KNIULINK Co., Ltd.**

NS69993 Design Notes for rpcs.cmn Digital Block

Version 1.0

February 25, 2022

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Ver. # | Rev. Date | Rev. By | Comment |
| 1.0 | 2022/2/25 | CaoHuiyang | Initial version |
| 1.1 | 2022/3/05 | CaoHuiyang | Add inter-structure and function |
| 1.2 |  |  |  |
| 1.3 |  |  |  |

**Contents**

[NS69993 Design Notes for rpcs.cmn Digital Block 0](#_Toc2807)

[1 Spec/Function Requirement/Function Description 3](#_Toc23403)

[2 Signal Definition 4](#_Toc29736)

[3 Implementation Details 11](#_Toc17558)

[3.1 Clock & Reset 11](#_Toc6288)

[3.2 Hardware Implementation 11](#_Toc19778)

[3.2.1 cmn\_creg 11](#_Toc23548)

[3.2.2 creg\_ctl 13](#_Toc11598)

[3.2.3 reg\_arbt 15](#_Toc29963)

[3.2.4 mem 16](#_Toc3629)

[3.2.5 mem\_arbt 17](#_Toc18281)

[3.2.6 mem\_reg0 18](#_Toc19645)

[3.2.7 mem\_reg1 18](#_Toc11483)

[3.2.8 mem\_reg2 18](#_Toc29344)

[3.2.9 mem\_reg3 18](#_Toc2980)

[3.2.10 reg\_mapping 19](#_Toc27539)

[3.3 Hardware summarize 20](#_Toc19718)

[4 User guide 21](#_Toc24157)

[4.1 Configuration Requirements 21](#_Toc7413)

[4.3 Registers 21](#_Toc3654)

[5 Special Note 21](#_Toc25652)

[6 FPGA Verification 21](#_Toc4410)

[7 Test Mode Strategy 21](#_Toc14443)

[8 Simulation List 21](#_Toc18429)

# Spec/Function Requirement/Function Description

* Register Management Mapping
* Lanes.fsm Memory and Memory Management
* Jtag/cr\_apb\_interface mux to cr\_interface

# Signal Definition

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Signal name** | | **I/O** | **From** | **To** | **Description/Function** |
| cr\_ext\_clk | | input | aon |  | cr外部时钟 |
| cr\_ext\_clk\_rst | | input |  | cr外部时钟复位 |
| cr\_clk | | input |  | cr时钟 |
| cr\_rst | | input |  | cr复位 |
| phy\_func\_reset | | output |  | aon |  |
|  | |  |  |  |  |
| ref\_range | | input | pcs\_raw |  |  |
| rx\_term\_offset[4:0] | | input | pcs\_raw |  | Rx电阻偏移 |
| txup\_term\_offset[8:0] | | input |  | Rx上拉电阻偏移 |
| txdn term\_offset[8:0] | | input |  | Rx下拉电阻偏移 |
| rtune\_req | | input |  | rtune请求 |
| fw\_pwrup done\_r | | output |  | aon | Send fw\_pwrup\_done\_r to AON cmn for MPLL force/skip logic |
|  | |  |  |  |  |
| fw\_rdy | | output |  | pcs\_raw\_lane | Generate fw\_rdy signal to lanes  一直拉高 |
| pma\_rx\_term\_offset[4:0] | | output |  | pcs\_raw->PMA | Rx电阻偏移给pma |
| pma txup\_term\_offset[8:0] | | output |  | Rx上拉电阻偏移给pma |
| pma txdn\_term\_offset[8:0] | | output |  | Rx下拉电阻偏移给pma |
|  | |  |  |  |  |
| pma\_rtune\_req | | output |  | pcs\_raw->PMA | rtune请求给pma |
|  | |  |  |  |  |
| cr\_para seli | | input | aon |  | from aon block--jtag and apb\_cr mux control signal(总线切换信号) |
| cr\_para\_addr[16:0] | | input | pcs\_raw |  | CREG parallel interface（cr总线）  (from apb or jtag) |
| cr\_para\_wr\_en | | input |  |
| cr\_para\_wr\_data[16:0] | | input |  |
| cr\_para\_rd\_en | | input |  |
| cr\_para\_rd\_data[16:0] | | output |  | pcs\_raw |
| cr\_para\_ack | | output |  |
| **Signal name** | | **I/O** | **From** | **To** | **Description/Function** |
| cr\_cmn\_aon sel[31:0] | | output |  | pcsraw\_aon\_cmn | aon register select control（aon模块寄存器译码后的片选） |
| cr\_cmn\_aon\_sel2[31:0] | | output |  |
| cr\_cmn\_aon\_rd\_data[15:0] | | input | pcsraw\_aon\_cmn |  | aon register read back data（aon模块寄存器回读数据） |
| Jtag\_trst\_n | input | | pcs\_raw |  | jtag复位 |
| jtag\_clk | input | | pcsraw\_aon\_cmn |  | jtag时钟 |
| jtag\_clk\_n | input | |  | jtag时钟反 |
| jtag\_tms | input | | pcs\_raw |  | jtag bus |
| jtag tdi | input | |  |
| jtag tdo | output | |  | pcs\_raw |
| jtag tdo\_en | output | |  |
| jtag\_apb\_sel | output | |  |  | 置1代表apb |
| ovrd\_clk\_sel | output | |  | aon |  |
| ovrd\_rst\_sel | output | |  |  |
| phy\_reset\_ov | output | |  |  |
| txrx0\_reset\_ov | output | |  |  |
| txrx1\_reset\_ov | output | |  |  |
| txrx2\_reset\_ov | output | |  |  |
| txrx3\_reset\_ov | output | |  |  |
| scan\_mode | input | | from pcs\_raw and input aon at the same time |  |  |
| scan\_shift | input | |  |  |
| scan shift\_cg | input | |  |  |
| scan\_set\_rst | input | |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| cr\_addr[15:0] | output |  | Raw PCS lane and PMA registers | CREG parallel interface (for Raw PCS lane and PMA registers)  (to Raw PCS lane and PMA) |
| cr\_wr\_en | output |  |
| cr\_wr\_data[15:0] | output |  |
| cr\_rd\_en | output |  |
|  |  |  |  |  |
| cr\_pma\_rd\_data[15:0] | input | PMA->pcs\_raw |  | CREG read data from PMA |
| **Signal name** | **I/O** | **From** | **To** | **Description/Function** |
| cr0\_rd\_data[15:0] | input | Raw PCS lanes |  | CREG read data from Raw PCS lanes(lane0-3,lane4-7 no use) |
| cr1\_rd\_data[15:0] | input |  |
| cr2\_rd\_data[15:0] | input |  |
| cr3\_rd\_data[15:0] | input |  |
| cr4\_rd\_data[15:0] | input |  |
| cr5\_rd\_data[15:0] | input |  |
| cr6\_rd\_data[15:0] | input |  |
| cr7\_rd\_data[15:0] | input |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| cr0\_mem\_ack | output |  | Raw PCS lanes | CREG interface with Memory-Arbiter((memory bus be used by lane.fsm,  lane0-3,lane4-7 no use)) |
| cr1\_mem\_ack | output |  |
| cr2\_mem\_ack | output |  |
| cr3\_mem\_ack | output |  |
| cr4\_mem\_ack | output |  |
| cr5\_mem\_ack | output |  |
| cr6\_mem\_ack | output |  |
| cr7\_mem\_ack | output |  |
|  |  |  |  |
| cr0\_mem\_rd\_data[15:0] | output |  | Raw PCS lanes |
| cr1\_mem\_rd\_data[15:0] | output |  |
| cr2\_mem\_rd\_data[15:0] | output |  |
| cr3\_mem\_rd\_data[15:0] | output |  |
| Cr4\_mem\_rd\_data[15:0] | output |  |
| cr5\_mem\_rd\_data[15:0] | output |  |
| cr6\_mem\_rd\_data[15:0] | output |  |
| cr7\_mem\_rd\_data[15:0] | output |  |
|  |  |  |  |
| cr0\_mem\_req | input | Raw PCS lanes |  |
| cr1\_mem\_req | input |  |
| cr2\_mem\_req | input |  |
| cr3\_mem\_req | input |  |
| cr4\_mem\_req | input |  |
| cr5\_mem\_req | input |  |
| cr6\_mem\_req | input |  |
| cr7\_mem\_req | input |  |
|  |  |  |  |
| cr0\_mem\_addr[15:0] | input | Raw PCS lanes |  |
| cr1\_mem\_addr[15:0] | input |  |
| cr2\_mem\_addr[15:0] | input |  |
| cr3\_mem\_addr[15:0] | input |  |
| cr4\_mem\_addr[15:0] | input |  |
| cr5\_mem\_addr[15:0] | input |  |
| cr6\_mem\_addr[15:0] | input |  |
| cr7\_mem\_addr[15:0] | input |  |
|  |  |  |  |
| cr0\_mem\_wr\_en | input | Raw PCS lanes |  |
| cr1\_mem\_wr\_en | input |  |
| cr2\_mem\_wr\_en | input |  |
| cr3\_mem\_wr\_en | input |  |
| cr4\_mem\_wr\_en | input |  |
| cr5\_mem\_wr\_en | input |  |
| cr6\_mem\_wr\_en | input |  |
| cr7\_mem\_wr\_en | input |  |
|  |  |  |  |
| cr0\_mem\_wr\_data[15:0] | input | Raw PCS lanes |  |
| cr1\_mem\_wr\_data[15:0] | input |  |
| cr2\_mem\_wr\_data[15:0] | input |  |
| cr3\_mem\_wr\_data[15:0] | input |  |
| cr4\_mem\_wr\_data[15:0] | input |  |
| cr5\_mem\_wr\_data[15:0] | input |  |
| cr6\_mem\_wr\_data[15:0] | input |  |
| cr7\_mem\_wr\_data[15:0] | input |  |
|  |  |  |  |
| cr0\_mem\_rd\_en | input | Raw PCS lanes |  |
| cr1\_mem\_rd\_en | input |  |
| cr2\_mem\_rd\_en | input |  |
| cr3\_mem\_rd\_en | input |  |
| cr4\_mem\_rd\_en | input |  |
| cr5\_mem\_rd\_en | input |  |
| cr6\_mem\_rd\_en | input |  |
| cr7\_mem\_rd\_en | input |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| cr0\_reg\_ack | output |  | Raw PCS lanes | CREG interface with Register-Arbiter  (此处译码产生的地址就是register Map address) |
| cr1\_reg\_ack | output |  |
| cr2\_reg ack | output |  |
| cr3\_reg\_ack | output |  |
| cr4\_reg\_ack | output |  |
| cr5\_reg\_ack | output |  |
| cr6\_reg\_ack | output |  |
| cr7\_reg\_ack | output |  |
|  |  |  |  |
| cr0\_reg\_rd\_data[15:0] | output |  | Raw PCS lanes |
| cr1\_reg\_rd\_data[15:0] | output |  |
| cr2\_reg\_rd\_data[15:0] | utput |  |
| cr3\_reg\_rd\_data[15:0] | output |  |
| cr4\_reg\_rd\_data[15:0] | output |  |
| cr5\_reg\_rd\_data[15:0] | output |  |
| cr6\_reg\_rd\_data[15:0] | output |  |
| cr7\_reg\_rd\_data[15:0] | output |  |
|  |  |  |  |
| cr0\_reg\_req | input | Raw PCS lanes |  |
| cr1\_reg\_req | input |  |
| cr2\_reg\_req | input |  |
| cr3\_reg\_req | input |  |
| cr4\_reg\_req | input |  |
| cr5\_reg\_req | input |  |
| cr6\_reg\_req | input |  |
| cr7\_reg\_req | input |  |
|  |  |  |  |
| cr0\_reg\_addr[15:0] | input | Raw PCS lanes |  |
| cr1\_reg\_addr[15:0] | input |  |
| cr2\_reg\_addr[15:0] | input |  |
| cr3\_reg addr[15:0] | input |  |
| cr4\_reg\_addr[15:0] | input |  |
| cr5\_reg\_addr[15:0] | input |  |
| cr6\_reg\_addr[15:0] | input |  |
| cr7\_reg\_addr[15:0] | input |  |
|  |  |  |  |
| cr0\_reg\_wr\_en | input | Raw PCS lanes |  |
| cr1\_reg\_wr\_en | input |  |
| cr2\_reg\_wr\_en | input |  |
| cr3\_reg\_wr\_en | input |  |
| cr4\_reg\_wr\_en | input |  |
| cr5\_reg\_wr\_en | input |  |
| cr6\_reg\_wr\_en | input |  |
| cr7\_reg\_wr\_en | input |  |
|  |  |  |  |
| cr0\_reg\_wr\_data[15:0] | input | Raw PCS lanes |  |
| cr1\_reg\_wr\_data[15:0] | input |  |
| cr2 reg\_wr\_data[15:0] | input |  |
| cr3\_reg\_wr\_data[15:0] | input |  |
| cr4\_reg\_wr\_data[15:0] | input |  |
| cr5\_reg\_wr\_data[15:0] | input |  |
| cr6\_reg\_wr\_data[15:0] | input |  |
| cr7\_reg\_wr\_data[15:0] | input |  |

# Implementation Details

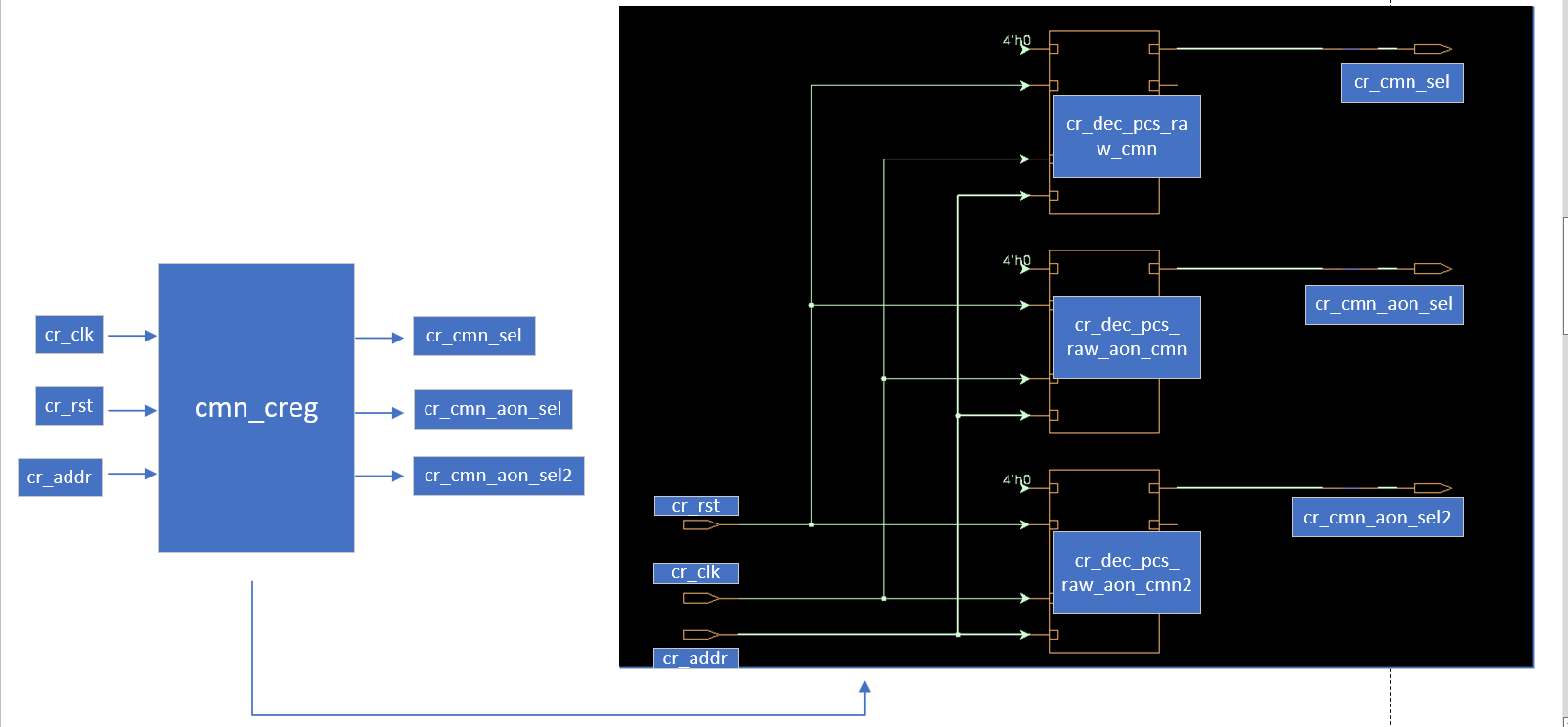
## Clock & Reset

cr\_ext\_clk产生自aon\_cmn根据cr\_para\_sel\_i切换自cr\_jtag\_clk或cr\_para\_clk，cr\_ext\_clk\_rst是cr\_ext\_clk同步了phy\_reset\_i。cr\_clk产生自cr\_ext\_clk。cr\_rst同步于cr\_clk产生自aon\_cmn由phy\_reset\_i和scan\_set\_rst产生。

## Hardware Implementation

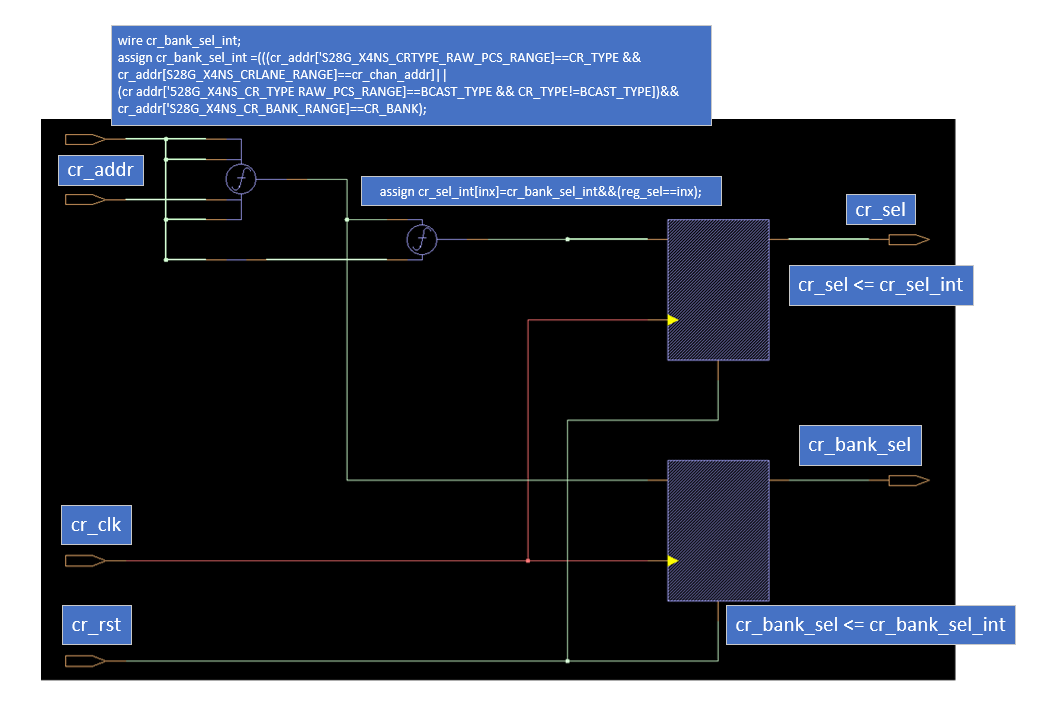
### cmn\_creg

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Signal name** | **I/O** | **From** | **To** | **Description/Function** |
| cr\_cmn\_sel[31:0] | output |  | pcs\_raw\_cmn | pcs\_raw\_cmn内部寄存器片选 |
| cr\_cmn\_aon\_sel[31:0] | output |  | pcs\_raw\_aon\_cmn | pcs\_raw\_aon\_cmn寄存器片选 |
| cr\_cmn\_aon\_sel2[31:0] | output |  | pcs\_raw\_aon\_cmn | pcs\_raw\_aon\_cmn寄存器片选 |
|  |  |  |  |  |
| cr\_clk | input | pcs\_raw\_cmn |  | cr时钟 |
| cr\_rst | input | pcs\_raw\_cmn |  | cr复位 |
| cr\_addr[15:0] | input | pcs\_raw\_cmn |  | cr地址总线(就是reg map地址) |



如上图所示，cmn\_creg模块内部包含了cr\_dec\_pcs\_raw\_cmn模块、cr\_dec\_pcs\_raw\_aon\_cmn模块和cr\_dec\_pcs\_raw\_aon\_cmn2模块。cmn\_creg模块的功能是用cr\_addr产生pcs\_raw\_cmn内部寄存器片选和pcs\_raw\_aon\_cmn寄存器片选。

cr\_dec\_pcs\_raw\_cmn 模块的功能是用cr\_addr分别译码产生cr\_cmn\_sel[31:0] 片选信号，cr\_dec\_pcs\_raw\_aon\_cmn 模块的功能是用cr\_addr分别译码产生cr\_cmn\_aon\_sel[31:0] 片选信号，cr\_dec\_pcs\_raw\_aon\_cmn2 模块的功能是用cr\_addr分别译码产生cr\_cmn\_aon\_sel2[31:0]片选信号。

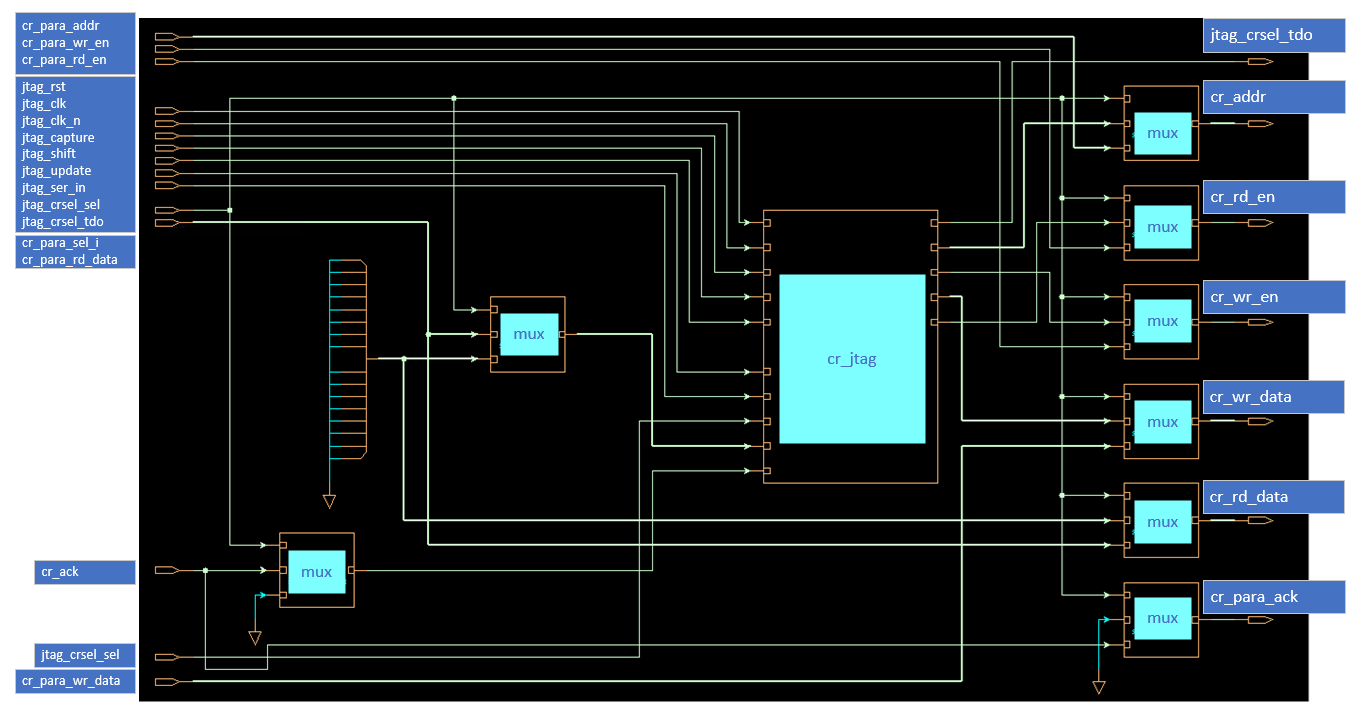


cr\_dec\_pcs\_raw\_cmn/cr\_dec\_pcs\_raw\_aon\_cmn/cr\_dec\_pcs\_raw\_aon\_cmn2内部结构相同，如上图所示，

cr\_addr经译码函数f1/f2锁存后产生cr\_sel和cr\_bank\_sel。其中cr\_bank\_sel\_int为寄存器bank片选信号，reg\_sel = cr\_addr[4:0]模块内部寄存器编号，cr\_sel\_int是reg\_sel独热码表示;cmn\_creg模块例化cr\_dec\_pcs\_raw\_cmn/cr\_dec\_pcs\_raw\_aon\_cmn/cr\_dec\_pcs\_raw\_aon\_cmn2模块时cr\_bank\_sel没用，而是直接通过三条总线将cr\_sel送到相应的模块。

### creg\_ctl

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Signal name** | **I/O** | **From** | **To** | **Description/Function** |
| cr\_para\_sel\_i | input | cmn |  | APB OR JTAG SELECT |
|  |  |  |  |  |
| cr\_para\_addr[15:0] | input | cmn |  | CR Parallel Interface |
| cr\_para\_wr\_en | input |  |
| cr\_para\_wr\_data[15:0] | input |  |
| cr\_para\_rd\_en | input |  |
| cr\_para\_rd\_data[15:0] | output |  | cmn |
| cr\_para\_ac | output |  |
|  |  |  |  |  |
| jtag\_rst | input | jtag\_ctl |  | JTAG Register interface |
| jtag\_clk | input |  |
| jtag\_clk\_n | input |  |
| jtag\_capture | input |  |
| jtag\_shift | input |  |
| jtag\_update | input |  |
| jtag\_ser\_in | input |  |
| jtag\_crsel\_sel | input |  |
| jtag\_crsel\_tdo | output |  | jtag\_ctl |
|  |  |  |  |  |
| cr\_addr[15:0] | output |  |  | Internal CREG interface |
| cr\_wr\_en | output |  |
| cr\_wr\_data[15:0] | output |  |
| cr\_rd\_en | output |  |
| cr\_rd\_data[15:0] | input |  |  |
| cr\_ack | input |  |  |



creg\_ctl 模块内部包含了creg\_jtag模块、cr\_addr\_mux模块、cr\_wr\_data\_mux 模块、cr\_wr\_en\_mux模块、cr\_rd\_en\_mux模块、cr\_para\_rd\_data\_mux模块、cr\_para\_ack\_mux模块、cr\_jtag\_rd\_data\_mux模块、cr\_jtag\_ack\_mux模块。cmn\_creg模块的功能是根据cr\_para\_sel\_i (默认CR Parallel Interface)来切换 CR Parallel Interface和JTAG Register interfac产生Internal CREG interface。

creg\_jtag模块的功能是将JTAG Register interface转换成cr interface。

cr\_addr\_mu模块是根据cr\_para\_sel\_i选择完成cr\_jtag\_addr和cr\_para\_addr到cr\_addr的切换。

cr\_wr\_data\_mux模块是根据cr\_para\_sel\_i选择完成cr\_jtag\_wr\_data和cr\_para\_wr\_data到cr\_wr\_data的切换。

cr\_wr\_en\_mux模块是根据cr\_para\_sel\_i选择完成cr\_jtag\_wr\_en和cr\_para\_wr\_en到cr\_wr\_en的切换。

cr\_rd\_en\_mux模块是根据cr\_para\_sel\_i选择完成cr\_jtag\_rd\_en和cr\_para\_rd\_en到cr\_rd\_en的切换。

cr\_para\_rd\_data\_mux模块是根据cr\_para\_sel\_i选择完成0和cr\_rd\_data到cr\_para\_rd\_data的切换。

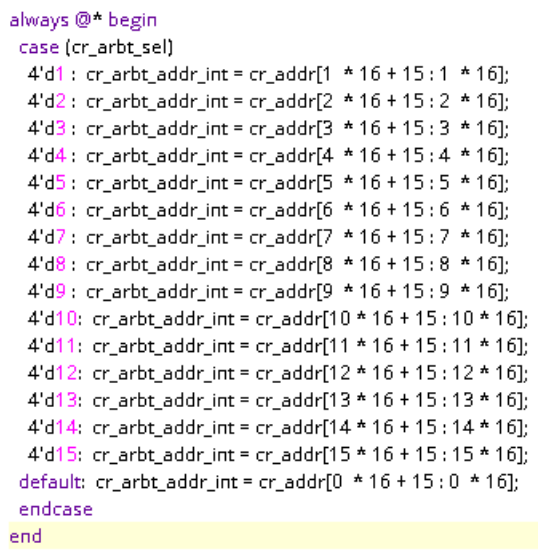
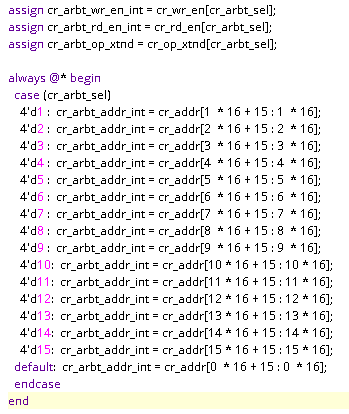
cr\_para\_ack\_mux模块是根据cr\_para\_sel\_i选择完成0和cr\_ack到cr\_para\_ack的切换。

cr\_jtag\_rd\_data\_mux模块是根据cr\_para\_sel\_i选择完成0和cr\_rd\_data到cr\_jtag\_rd\_data的切换。

cr\_jtag\_ack\_mux模块是根据cr\_para\_sel\_i选择完成0和cr\_ack到cr\_jtag\_ack的切换。

### reg\_arbt

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Signal name** | **I/O** | **From** | **To** | **Description/Function** |
| cr\_clk | input |  |  | Clock and Reset |
| cr\_rst | input |  |  |
|  |  |  |  |  |
| cr\_req[15:0] | input |  |  | CR input interfaces （将16路cr interface 合并成256位总线输入，此处的地址就是reg map地址） |
| cr\_addr[255:0] | input |  |  |
| cr\_wr\_data[255:0] | input |  |  |
| cr\_wr\_en[15:0] | input |  |  |
| cr\_rd\_en[15:0] | input |  |  |
| cr\_op\_xtnd[15:0] | input |  |  |
| cr\_ack[15:0] | output |  |  |
| cr\_rd data[15:0] | output |  |  |
|  |  |  |  |  |
| cr\_arbt\_addr[15:0] | output |  |  | CR output interface（切换后的cr\_interface输出，此处的地址就是reg map地址） |
| cr\_arbt\_wr\_data[15:0] | output |  |  |
| cr\_arbt\_wr\_en | output |  |  |
| cr\_arbtrd\_en | output |  |  |
| cr\_arbt rd\_data[15:0] | input |  |  |

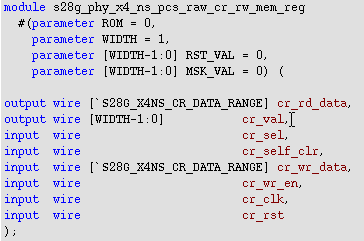


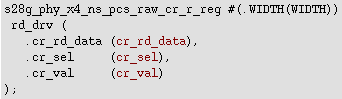
如上所示，reg\_arbt模块的功能是根据cr\_arbt\_sel的值决定16路CR input interfaces的哪一路作为CR output interface，cr\_arbt\_sel由input wire [16-1:0] cr\_req译码产生。本设计中用到4路，所以reg\_arbt模块实现了4路cr interface 并行输入转换为4拍串行1路CR output interface输出的过程。

### mem

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Signal name** | **I/O** | **From** | **To** | **Description/Function** |
| cr\_c1k | input |  |  | mem时钟 |
| cr\_rst | input |  |  | mem复位 |
|  |  |  |  |  |
| cr\_cmn[63:0]\_b[7:0]\_sel[31:0] | input | mem\_creg[3:0] |  | mem中register片选 |
|  |  |  |  |  |
| cr\_wr\_en | input |  |  | mem中register写使能 |
| cr\_wr\_data[15:0] | input |  |  | mem中register写数据 |
| cr\_rd\_en | input |  |  | mem中register读使能 |
| cr\_rd data[15:0] | output |  | Raw PCS lanes | mem中register读数据 |

mem用cmn[63:0]\_b[7:0],b[7:0]表示8个bank，cmn[63:0]表示64个block，每个cmn block中包含32个16位的register，所以如果用4个bank，memory的大小为64（block）\*32（register）\*4（bank）=8192 registers (16bit)。mem的存储单元是寄存器接口和具体操作如下。





### mem\_arbt

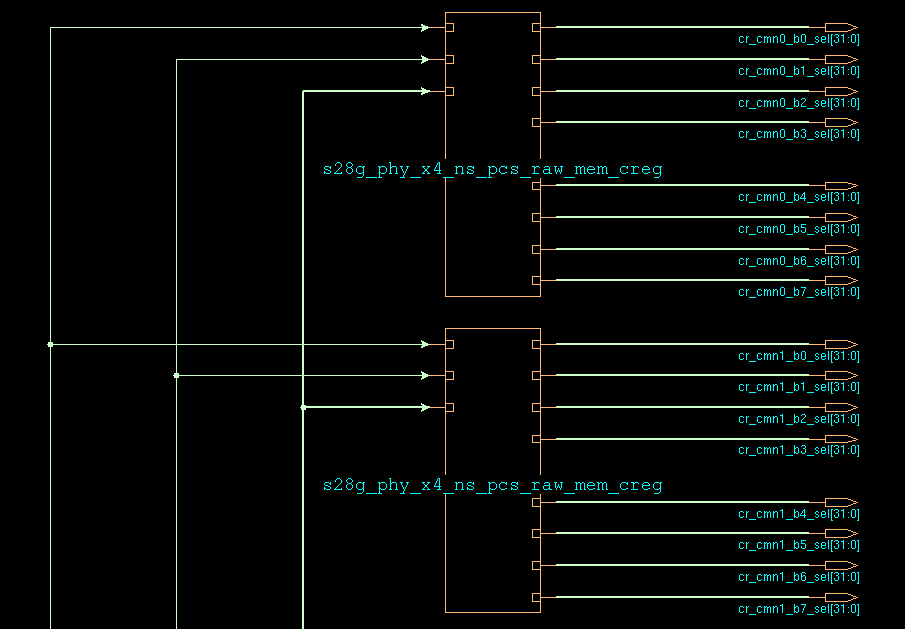
mem\_arb模块接口与reg\_arbt模块相同，mem\_arb模块功能与reg\_arbt模块相同。

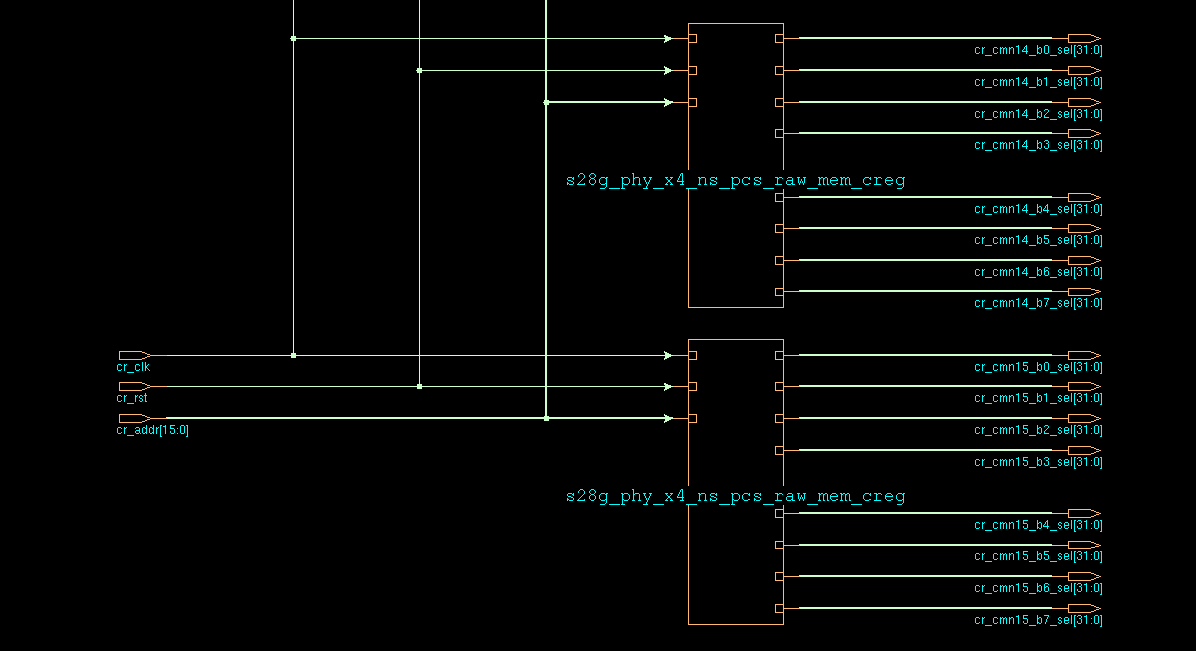
区别在于mem\_arb的输入输出为mem的地址和数据。

### mem\_reg0

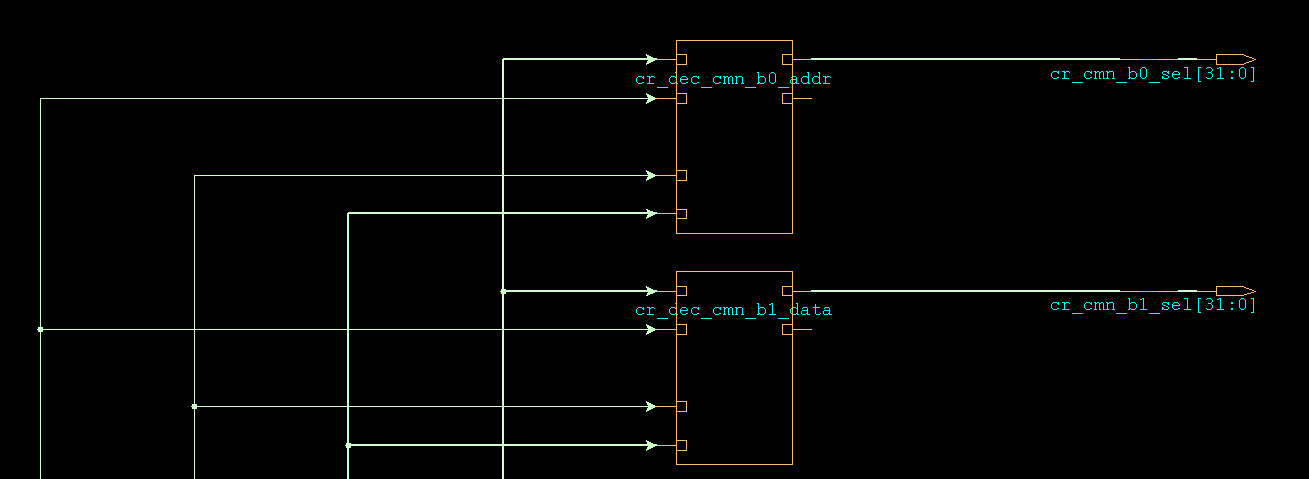
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Signal name** | **I/O** | **From** | **To** | **Description/Function** |
| cr\_ck | input |  |  | mem时钟 |
| cr\_rst | input |  |  | mem复位 |
| cr\_addr | input | Raw PCS lanes |  | mem地址 |
|  |  |  |  |  |
| cr\_cmn[15:0]\_b[7:0]\_sel | output |  | mem | mem中register片选 |

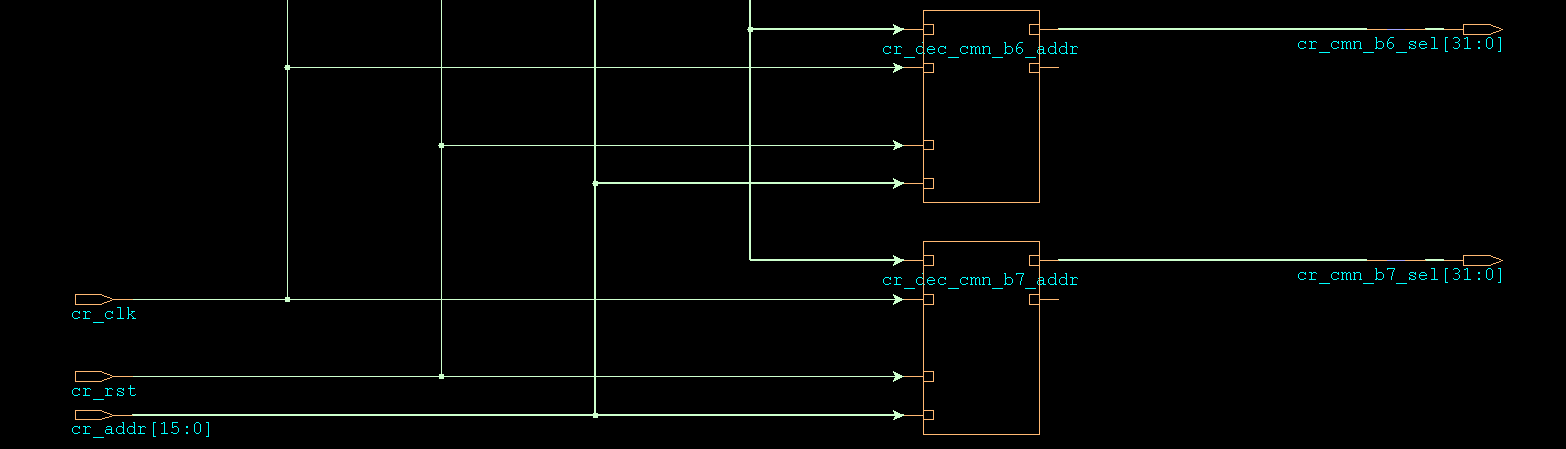
mem\_reg模块的功能是通过对cr\_addr[15:0](cr\_addr外部连接cr\_mem\_addr)的译码产生cmn[15:0]\_b[7:0][31:0]的片选信号。





上图为cmn0\_creg~cmn15\_creg((block选择信号产生)),下图为cmnX\_creg(bank选择信号产生)内部结构





cr\_dec\_cmn\_bx\_addr模块的内部结构及实现方法与cr\_dec\_pcs\_raw\_cmn/cr\_dec\_pcs\_raw\_aon\_cmn/cr\_dec\_pcs\_raw\_aon\_cmn2内部结构及实现方法相同。

### mem\_reg1

接口/内部结构及实现方法同上

产生cmn[31:16]\_b[7:0]的片选信号

### mem\_reg2

接口/内部结构及实现方法同上

产生cmn[47:32]\_b[7:0]的片选信号

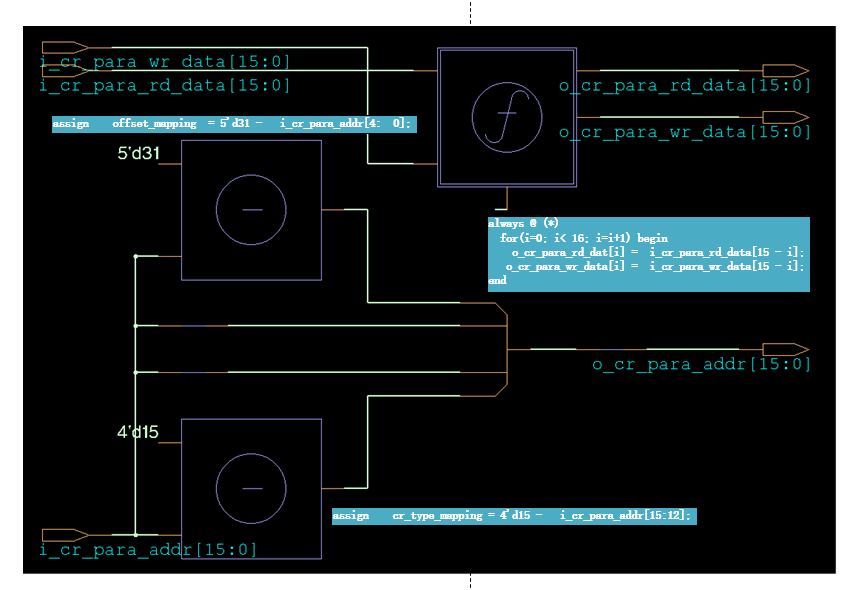
### mem\_reg3

接口/内部结构及实现方法同上

产生cmn[63:48]\_b[7:0]的片选信号

### reg\_mapping

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Signal name** | **I/O** | **From** | **To** | **Description/Function** |
| i\_cr\_para\_addr[15:0] | input |  |  | cr\_para\_add before mapping |
| i\_cr para\_rd data[15:0] | input |  |  | cr para\_rd data before mapping |
| i\_cr para wr\_data[15:0] | input |  |  | cr para wr\_data before mapping |
|  |  |  |  |  |
| o\_cr para addr[15:0] | output |  |  | cr\_para\_add after mapping |
| o\_cr para rd data[15:0] | output |  |  | cr para\_rd data after mapping |
| o\_cr para\_wr\_data[15:0] | output |  |  | cr para wr\_data after mapping |



如上图所示，reg\_mapping主要功能是完成对cr\_para\_add/cr\_para\_rd\_data/cr\_para\_wr\_data的映射操作。

## Hardware summarize

cr\_ para\_ addr[15:12]=4'b0101,PHY broadcasts to all PMA lanes.

cr\_para\_ addr[15:12]=4b0110, PHY broadcasts to all RAW PCS lanes.

cr\_para\_ addr[15:12]=4b0111, PHY broadcasts to all RAW PCS AON lanes.

The registers can be operated by firmware/jtag/apb.

Frimware operate register flow:1.Xlane.fsm get order and operate data through cr[7:0]\_mem\_bus ❶ ;2. Xlane.fsm complete register operate with reg\_arbt through cr[7:0]\_reg\_bus❷;2. reg\_arbt complete register operate with registers in every module through cr\_bus❸ ;



# User guide

## 4.1 Configuration Requirements

## 4.3 Registers

# Special Note

# FPGA Verification

# Test Mode Strategy

# Simulation List