
Mistral documentation

Release 1.0

Dec 30, 2020

CONTENTS

1	The Cyclone V FPGA	1
1.1	The FPGAs	1
1.2	Bitstream stucture	2
1.3	Logic blocks	2
1.4	Routing network	3
1.5	Programmable inverters	3
2	CycloneV internals description	5
2.1	Routing network	5
2.2	Inner logic blocks	6
2.3	Peripheral logic blocks	19
2.4	Options	92
3	CycloneV library usage	93
3.1	Library structure	93
3.2	Packages	93
3.3	Model information	94
3.4	pos, rnode and pnode	95
3.5	Routing network management	96
3.6	Logic block management	97
3.7	Inverters management	98
3.8	Pin/package management	98
3.9	Options	99
3.10	Bitstream management	100
4	The mistral-cv command-line program	101
4.1	models	101
4.2	routes	101
4.3	routes2	101
4.4	cycle	101
4.5	bels	102
4.6	decomp	102
4.7	comp	102
4.8	diff	102
5	Mistral CycloneV library internals	103
5.1	Structure	103
5.2	Routing data	103
5.3	Block muxes	104
5.4	Logic blocks	105

5.5	Inverters	105
5.6	Forced-1 bits	105
5.7	Packages	105
5.8	Models	105

THE CYCLONE V FPGA

1.1 The FPGAs

The Cyclone V is a series of FPGAs produced initially by Altera, now Intel. It is based on a series of seven dies with varying levels of capability, which is then derived into more than 400 SKUs with variations in speed, temperature range, and enabled internal hardware.

As pretty much every FPGA out there, the dies are organized in grids.

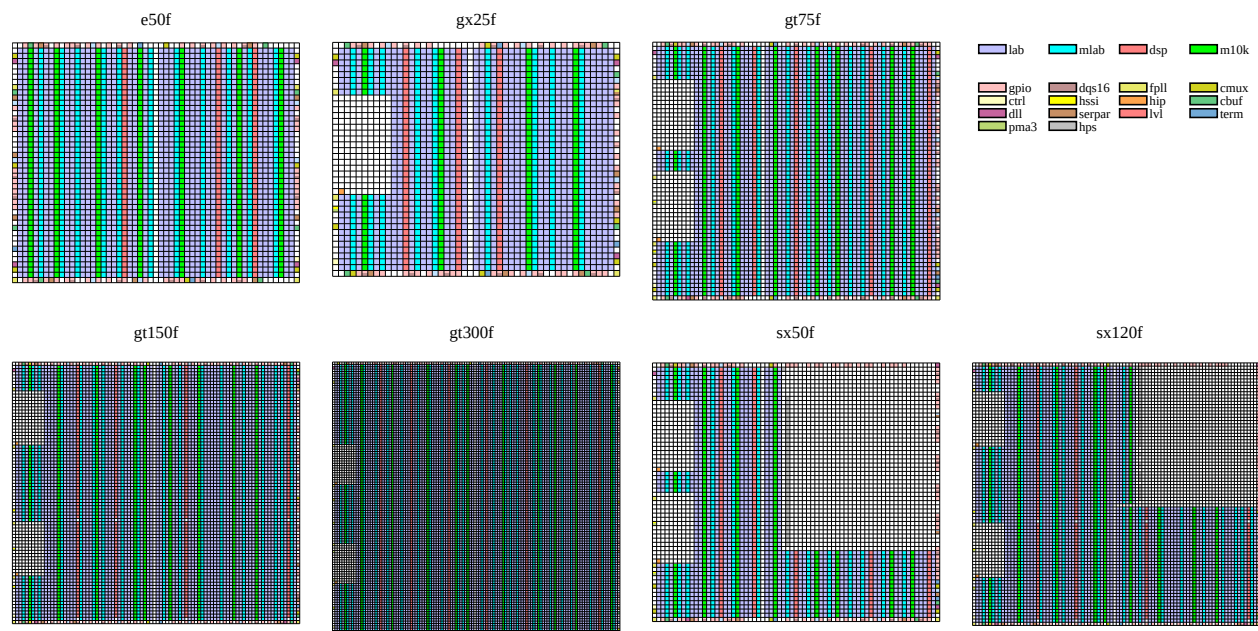


Fig. 1: Floor plan of the seven die types

The FPGA, structurally, is a set of logic blocks of different types communicating with each other either through direct links or through a large routing network that spans the whole grid.

Some of the logic blocks take visible floor space. Specifically, the notches on the left are the space taken by the high speed serial interfaces (hssi and pma3). Also, the top-right corner in the sx50f and sx120f variants is used to fit the hps, a dual-core arm.

1.2 Bitstream stucture

The bitstream is built from three rams:

- Option ram
- Peripheral ram
- Configuration ram

The option ram is composed of 32 blocks of 40 bits, of which only 12 are actually used. It includes the global configurations for the chip, such as the jtag user id, the programming voltage, the internal oscillator configuration, etc.

The peripheral ram stores the configuration of all the blocks situated on the borders of the chip, e.g. everything outside of labs, mlabs, dsps and m10ks. It is built of 13 to 16 blocks of bits that are sent through shift registers to the tiles.

The configuration ram stores the configuration of the labs, mlabs, dsps and m10ks, plus all the routing configuration. It also includes the programmable inverters which allows inverting essentially all the inputs to the peripheral blocks. It is organised as a rectangle of bits.

Die	Tiles	Pram	Cram
e50f	55x46	51101	4958x3928
gx25f	49x40	54083	3856x3412
gt75f	69x62	90162	6006x5304
gt150f	90x82	113922	7605x7024
gt300f	122x116	130828	10038x9948
sx50f	69x62	80505	6006x5304
sx120f	90x82	99574	7605x7024

1.3 Logic blocks

The logic blocks are of two categories, the inner blocks and the peripheral blocks. To a first approximation all the inner blocks are configured through configuration ram, and the peripheral blocks through the peripheral ram. It only matters where it comes to partial reconfiguration, because only the configuration ram can be dynamically modified. We do not yet support it though.

The inner blocks are:

- lab: a logic blocks group with 20 LUTs with 5 inputs and 40 Flip-Flops.
- mlab: a lab that can be reconfigured as 64*20 bits of ram
- dsp: a flexible multiply-add block
- m10k: a block of 10240 bits of dual-ported memory

The peripheral blocks are:

- gpio: general-purpose i/o, a block that controls up to 4 package pins
- dqs16: a block that manage differential input/output for 4 gpio blocks, e.g. up to 16 pins
- fppll: a fractional PLL
- cmux: the clock muxes that drive the clock part of the routing network
- ctrl: the control block with things like jtag
- hssi: the high speed serial interfaces

- hip: the pcie interfaces
- cbuf: a clock buffer for the dqs16
- dll: a delay-locked loop for the dqs16
- serpar: TODO
- lvl: TODO
- term: termination control blocks
- pma3: manages the channels of the hssi
- hmc: hardware memory controller, a block managing sdr/ddr ram interfaces
- hps: a series of 37 blocks managing the interface with the integrated dual-core arm

All of these blocks are configured similarly, through the setup of block muxes. They can be of 4 types: * Boolean
* Symbolic, where the choice is between alphanumeric states * Numeric, where the choice is between a fixed set of numeric value * Ram, where a series of bits can be set to any value

Configuring that part of the FPGA consists of configuring the muxes associated to each block.

1.4 Routing network

A massive routing network is present all over the FPGA. It has two almost-disjoint parts. The data network has a series of inputs, connected to the outputs of all the blocks, and a series of outputs that go to data inputs of the blocks. The clock network consists of 16 global clocks signals that cover the whole FPGA, up to 88 regional clocks that cover an half of the FPGA, and when an hssi is present a series of horizontal peripheral clocks that are driven by the serial communications. Global and regional clock signals are driven by dedicated cmux blocks (not the fppl in particular, but they do have dedicated connections to the cmuxes).

These two networks join on data/clock muxes, which allow peripheral blocks to select for their clock-like inputs which network the signal should come from.

1.5 Programmable inverters

Essentially every output of the routing network that enters a peripheral block can optionally be inverted by activating the associated configuration bit.

CYCLONEV INTERNALS DESCRIPTION

2.1 Routing network

The routing network follows a single-driver structure: a number of inputs are grouped together in one place, one is selected through the configuration, then it is amplified and used to drive a metal line. There is also usually one bit configuration to disable the driver, which can be all-off (probably leaving the line floating) or a specific combination to select vcc. The drivers correspond to a 2d pattern in the configuration ram. There are 70 different patterns, configured by 1 to 18 bits and mixing 1 to 44 inputs.

The network itself can be split in two parts: the data network and the clock network.

The data network is a grid of connections. Horizontal lines (H14, H6 and H3, numbered by the number of tiles they span) and vertical lines (V12, V4 and V2) helped by wire muxes (WM) connect to each other to ensure routing over the whole surface. Then at the tile level tile-data dispatch (TD) nodes allow to select between the available signals.

Generic output (GOUT) nodes then select between TD nodes to connect to logic blocks inputs. Logic block outputs go to Generic Input (GIN) nodes which feed in the connections. In addition a dedicated network, the Loopback dispatch (LD) connects some of the outputs from the labs/mlabs to their inputs for fast local data routing.

The clock network is more of a top-down structure. The top structures are Global clocks (GCLK), Regional clocks (RCLK) and Peripheral clocks (PCLK). They're all driven by specialized logic blocks we call Clock Muxes (cmux). There are two horizontal cmux in the middle of the top and bottom borders, each driving 4 GCLK and 20 RCLK, two vertical in the middle of the left and right borders each driving 4 GCLK and 12 RCLK, and 3 to 4 in the corners driving 6 RCLK each. The dies including an HPS (sx50f and sx120f) are missing the top-right cmux plus some of the middle-of-border-driven RCLK. That gives a total of 16 GCLK and 66 to 88 RCLK. In addition PCLK start from HSSI blocks to distribute serial clocks to the network.

The GCLK span the whole grid. A RCLK spans half the grid. A PCLK spans a number of tiles horizontally to its right.

The second level is Sector clocks, SCLK, which spans small rectangular zones of tiles and connect from GCLK, RCLK and PCLK. The on the third level, connecting from SCLK, is Horizontal clocks (HCLK) spanning 10-15 horizontal tiles and Border clocks (BCLK) rooted regularly on the top and bottom borders. Finally Tile clocks (TCLK) connect from HCLK and BCLK and distribute the clocks within a tile.

In addition the PMUX nodes at the entrance of plls select between SCLKs, and the GCLKFB and RCLKFB bring back feedback signals from the cmux to the pll.

Inner blocks directly connect to TCLK and have internal muxes to select between clock and data inputs for their control. Peripheral blocks tend to use a secondary structure composed from a TDMUX that selects one TD between multiple ones followed by a DCMUX that selects between the TDMUX and a TCLK so that their clock-like inputs can be driven from either a clock or a data signal.

Most GOUT and DCMUX connected to inputs to peripheral blocks are also provided with an optional inverter.

2.2 Inner logic blocks

2.2.1 LAB

The LABs are the main combinatorial and register blocks of the FPGA. A LAB tile includes 10 sub-blocks with 64 bits of LUT splitted in 6 parts, four Flip-Flops, two 1-bit adders and a lot of routing logic. In addition a common control subblock selects and dispatches clock, enable, clear, etc signals.

Name	Instance	Type	Values	Default	Documentation
ARITH_SEL	0-9	Mux	<ul style="list-style-type: none"> • adder • lut 	lut	TODO
BCLK_SEL	0-9	Mux	<ul style="list-style-type: none"> • off • clk0 • clk1 • clk2 	off	TODO
BCLR_SEL	0-9	Num	<ul style="list-style-type: none"> • 0-1 	0	TODO
BDFF0	0-9	Mux	<ul style="list-style-type: none"> • reg • nlut 	reg	TODO
BDFF1	0-9	Mux	<ul style="list-style-type: none"> • reg • nlut 	reg	TODO
BDFF1L	0-9	Mux	<ul style="list-style-type: none"> • reg • nlut 	reg	TODO
BEF_SEL	0-9	Mux	<ul style="list-style-type: none"> • e • f 	e	TODO
BPKREG0	0-9	Bool	t/f	f	TODO
BPKREG1	0-9	Bool	t/f	f	TODO
BSCLR_DIS	0-9	Bool	t/f	f	TODO
BSLOAD_EN	0-9	Bool	t/f	f	TODO
B_FEEDBACK_SEL	0-9	Num	<ul style="list-style-type: none"> • 0-1 	0	TODO
LUT_MASK	0-9	Ram	64 bits	0	TODO

continues on next page

Table 1 – continued from previous page

Name	Instance	Type	Values	Default	Documentation
MODE	0-9	Mux	<ul style="list-style-type: none"> • l5 • l5_ft • l5_fb • l5_ftb • l6 • l6_ft • l6_fb • l6_ftb • l7_e0 • l7_e0_ft • l7_e0_fb • l7_e0_ftb • l7_e1 • l7_e1_ft • l7_e1_fb • l7_e1_ftb 	l6	TODO
SHARE	0-9	Bool	t/f	f	TODO
TCLK_SEL	0-9	Mux	<ul style="list-style-type: none"> • off • clk0 • clk1 • clk2 	off	TODO
TCLR_SEL	0-9	Num	<ul style="list-style-type: none"> • 0-1 	0	TODO
TDFF0	0-9	Mux	<ul style="list-style-type: none"> • reg • nlut 	reg	TODO
TDFF1	0-9	Mux	<ul style="list-style-type: none"> • reg • nlut 	reg	TODO
TDFF1L	0-9	Mux	<ul style="list-style-type: none"> • reg • nlut 	reg	TODO
TEF_SEL	0-9	Mux	<ul style="list-style-type: none"> • e • f 	e	TODO
TPKREG0	0-9	Bool	t/f	f	TODO
TPKREG1	0-9	Bool	t/f	f	TODO
TSCLR_DIS	0-9	Bool	t/f	f	TODO
TSLOAD_EN	0-9	Bool	t/f	f	TODO

continues on next page

Table 1 – continued from previous page

Name	Instance	Type	Values	Default	Documentation
T_FEEDBACK_SEL9		Num	<ul style="list-style-type: none"> 0-1 	0	TODO
ACLR0_INV		Bool	t/f	f	TODO
ACLR0_SEL		Mux	<ul style="list-style-type: none"> gin1 clki2 	gin1	TODO
ACLR1_INV		Bool	t/f	f	TODO
ACLR1_SEL		Mux	<ul style="list-style-type: none"> gin0 clki3 	gin0	TODO
BTO_DIS		Bool	t/f	f	TODO
BYPASS_DIS		Bool	t/f	t	TODO
CLK0_INV		Bool	t/f	f	TODO
CLK0_SEL		Mux	<ul style="list-style-type: none"> clka clkb 	clka	TODO
CLK1_INV		Bool	t/f	f	TODO
CLK1_SEL		Mux	<ul style="list-style-type: none"> clka clkb 	clka	TODO
CLK2_INV		Bool	t/f	f	TODO
CLK2_SEL		Mux	<ul style="list-style-type: none"> clka clkb 	clka	TODO
CLKA_SEL		Mux	<ul style="list-style-type: none"> clki0 gin2 	clki0	TODO
CLKB_SEL		Mux	<ul style="list-style-type: none"> clki1 gin3 	clki1	TODO
DFT_MODE		Mux	<ul style="list-style-type: none"> off on dft_pprog 	on	TODO
EN0_EN		Bool	t/f	t	TODO
EN0_NINV		Bool	t/f	t	TODO
EN0_SEL		Mux	<ul style="list-style-type: none"> gin1 gin3 	gin1	TODO
EN1_EN		Bool	t/f	t	TODO

continues on next page

Table 1 – continued from previous page

Name	Instance	Type	Values	Default	Documentation
EN1_NINV		Bool	t/f	t	TODO
EN1_SEL		Mux	<ul style="list-style-type: none"> • gin0 • gin3 	gin3	TODO
EN2_EN		Bool	t/f	t	TODO
EN2_NINV		Bool	t/f	t	TODO
EN_SCLK_LOAD_WHAT		Bool	t/f	f	TODO
REGSCAN_LATCH_EN		Bool	t/f	f	TODO
SCLR_INV		Bool	t/f	f	TODO
SCLR_MUX		Mux	<ul style="list-style-type: none"> • gin3 • gin2 	gin3	TODO
SLOAD_INV		Bool	t/f	t	TODO
SLOAD_SEL		Mux	<ul style="list-style-type: none"> • gin0 • gin3 	gin0	TODO
TTO_DIS		Bool	t/f	f	TODO

2.2.2 MLAB

A MLAB is a lab that can optionally be turned into a 640-bits RAM or ROM. The wiring is identical to the LAB, only some additional muxes are provided to select the RAM/ROM mode.

TODO: address/data wiring in RAM/ROM mode.

Name	Instance	Type	Values	Default	Documentation
ACLR0_INV		Bool	t/f	f	TODO
ACLR0_SEL		Mux	<ul style="list-style-type: none"> • gin1 • clki2 	gin1	TODO
ACLR1_INV		Bool	t/f	f	TODO
ACLR1_SEL		Mux	<ul style="list-style-type: none"> • gin0 • clki3 	gin0	TODO
BTO_DIS		Bool	t/f	f	TODO
BYPASS_DIS		Bool	t/f	t	TODO
CLK0_INV		Bool	t/f	f	TODO
CLK0_SEL		Mux	<ul style="list-style-type: none"> • clka • clk b 	clka	TODO
CLK1_INV		Bool	t/f	f	TODO

continues on next page

Table 2 – continued from previous page

Name	Instance	Type	Values	Default	Documentation
CLK1_SEL		Mux	<ul style="list-style-type: none"> • clka • clkb 	clka	TODO
CLK2_INV		Bool	t/f	f	TODO
CLK2_SEL		Mux	<ul style="list-style-type: none"> • clka • clkb 	clka	TODO
CLKA_SEL		Mux	<ul style="list-style-type: none"> • clki0 • gin2 	clki0	TODO
CLKB_SEL		Mux	<ul style="list-style-type: none"> • clki1 • gin3 	clki1	TODO
DFT_MODE		Mux	<ul style="list-style-type: none"> • off • on • dft_pprog 	on	TODO
EN0_EN		Bool	t/f	t	TODO
EN0_NINV		Bool	t/f	t	TODO
EN0_SEL		Mux	<ul style="list-style-type: none"> • gin1 • gin3 	gin1	TODO
EN1_EN		Bool	t/f	t	TODO
EN1_NINV		Bool	t/f	t	TODO
EN1_SEL		Mux	<ul style="list-style-type: none"> • gin0 • gin3 	gin3	TODO
EN2_EN		Bool	t/f	t	TODO
EN2_NINV		Bool	t/f	t	TODO
EN_SCLK_LOAD_WHAT		Bool	t/f	f	TODO
MADDG_VOLTAGE		Mux	<ul style="list-style-type: none"> • vccl • vcchg 	vccl	TODO
MCRG_VOLTAGE		Mux	<ul style="list-style-type: none"> • vcchg • vccl 	vcchg	TODO
RAM_DIS		Bool	t/f	t	TODO
REGSCAN_LATCH_EN		Bool	t/f	f	TODO
SCLR_INV		Bool	t/f	f	TODO

continues on next page

Table 2 – continued from previous page

Name	Instance	Type	Values	Default	Documentation
SCLR_MUX		Mux	<ul style="list-style-type: none"> • gin3 • gin2 	gin3	TODO
SLOAD_INV		Bool	t/f	t	TODO
SLOAD_SEL		Mux	<ul style="list-style-type: none"> • gin0 • gin3 	gin0	TODO
TTO_DIS		Bool	t/f	f	TODO
WRITE_EN		Bool	t/f	f	TODO
WRITE_PULSE_LENGTH		Num	<ul style="list-style-type: none"> • 500 • 650 • 800 • 950 	500	TODO
ARITH_SEL	0-9	Mux	<ul style="list-style-type: none"> • adder • lut 	lut	TODO
BCLK_SEL	0-9	Mux	<ul style="list-style-type: none"> • off • clk0 • clk1 • clk2 	off	TODO
BCLR_SEL	0-9	Num	<ul style="list-style-type: none"> • 0-1 	0	TODO
BDFF0	0-9	Mux	<ul style="list-style-type: none"> • reg • nlut 	reg	TODO
BDFF1	0-9	Mux	<ul style="list-style-type: none"> • reg • nlut 	reg	TODO
BDFF1L	0-9	Mux	<ul style="list-style-type: none"> • reg • nlut 	reg	TODO
BEF_SEL	0-9	Mux	<ul style="list-style-type: none"> • e • f 	e	TODO
BPKREG0	0-9	Bool	t/f	f	TODO
BPKREG1	0-9	Bool	t/f	f	TODO
BSCLR_DIS	0-9	Bool	t/f	f	TODO
BSLOAD_EN	0-9	Bool	t/f	f	TODO

continues on next page

Table 2 – continued from previous page

Name	Instance	Type	Values	Default	Documentation
B_FEEDBACK_SEL	0-9	Num	<ul style="list-style-type: none"> • 0-1 	0	TODO
LUT_MASK	0-9	Ram	64 bits	0	TODO
MODE	0-9	Mux	<ul style="list-style-type: none"> • 15 • 15_ft • 15_fb • 15_ftb • 16 • 16_ft • 16_fb • 16_ftb • 17_e0 • 17_e0_ft • 17_e0_fb • 17_e0_ftb • 17_e1 • 17_e1_ft • 17_e1_fb • 17_e1_ftb 	16	TODO
SHARE	0-9	Bool	t/f	f	TODO
TCLK_SEL	0-9	Mux	<ul style="list-style-type: none"> • off • clk0 • clk1 • clk2 	off	TODO
TCLR_SEL	0-9	Num	<ul style="list-style-type: none"> • 0-1 	0	TODO
TDF0	0-9	Mux	<ul style="list-style-type: none"> • reg • nlut 	reg	TODO
TDF1	0-9	Mux	<ul style="list-style-type: none"> • reg • nlut 	reg	TODO
TDF1L	0-9	Mux	<ul style="list-style-type: none"> • reg • nlut 	reg	TODO
TEF_SEL	0-9	Mux	<ul style="list-style-type: none"> • e • f 	e	TODO
TPKREG0	0-9	Bool	t/f	f	TODO
TPKREG1	0-9	Bool	t/f	f	TODO

continues on next page

Table 2 – continued from previous page

Name	Instance	Type	Values	Default	Documentation
TSCLR_DIS	0-9	Bool	t/f	f	TODO
TSLOAD_EN	0-9	Bool	t/f	f	TODO
T_FEEDBACK_SEL	0-9	Num	• 0-1	0	TODO

2.2.3 DSP

The DSP blocks provide a multiply-adder with either three 9x9, two 18x18 or one 27x27 multiply, and the 64-bits accumulator. Its large number of inputs and output makes it span two tiles vertically.

TODO: everything, GOUT/GIN/DCMUX mapping is done

Name	Type	Values	Default	Documentation
ACC_INV	Bool	t/f	f	TODO
AX_SIGNED	Bool	t/f	f	TODO
AY_SIGNED	Bool	t/f	f	TODO
BX_SIGNED	Bool	t/f	f	TODO
BY_SIGNED	Bool	t/f	f	TODO
CASCADE_1ST_EN	Bool	t/f	f	TODO
CASCADE_EN	Bool	t/f	f	TODO
CE_SMUX0_FORCE	Bool	t/f	f	TODO
CE_SMUX0_INV	Bool	t/f	f	TODO
CE_SMUX1_FORCE	Bool	t/f	f	TODO
CE_SMUX1_INV	Bool	t/f	f	TODO
CE_SMUX2_FORCE	Bool	t/f	f	TODO
CE_SMUX2_INV	Bool	t/f	f	TODO
CHAIN_OUTPUT_EN	Bool	t/f	f	TODO
CLK_AX17_SEL	Num	• 0-2	0	TODO
CLK_AYZ17_SEL	Num	• 0-2	0	TODO
CLK_BX17_SEL	Num	• 0-2	0	TODO
CLK_BYZ17_SEL	Num	• 0-2	0	TODO
CLK_DYN_CTRL_SEL	Num	• 0-2	0	TODO
CLK_OPREG_SEL	Num	• 0-2	0	TODO
CLK_SMUX0_INV	Bool	t/f	f	TODO
CLK_SMUX0_INV	Bool	t/f	f	TODO

continues on next page

Table 3 – continued from previous page

Name	Type	Values	Default	Documentation
CLK_SMUX0_SEL	Mux	<ul style="list-style-type: none"> labclk0 lsim6 	labclk0	TODO
CLK_SMUX1_SEL	Mux	<ul style="list-style-type: none"> labclk1 lsim8 	labclk1	TODO
CLK_SMUX2_INV	Bool	t/f	f	TODO
CLK_SMUX2_SEL	Mux	<ul style="list-style-type: none"> labclk2 lsim0 	labclk2	TODO
COEF_H	Ram	144 bits	0	TODO
COEF_INPUT_EN	Bool	t/f	f	TODO
COEF_L	Ram	144 bits	0	TODO
DEC_INV	Bool	t/f	f	TODO
DELAY_CASCADE_A64_EN	Bool	t/f	f	TODO
DELAY_CASCADE_B64_EN	Bool	t/f	f	TODO
DFT_CLK_DIS	Bool	t/f	t	TODO
DFT_ITG_EN	Bool	t/f	f	TODO
DFT_TDF_EN	Bool	t/f	f	TODO
DOUBLE_ACC_EN	Bool	t/f	f	TODO
IDIREG_ACC_CTRL	Mux	<ul style="list-style-type: none"> bypass reg 	bypass	TODO
IDIREG_DEC_CTRL	Mux	<ul style="list-style-type: none"> bypass reg 	bypass	TODO
IDIREG_PRELOAD_CTRL	Mux	<ul style="list-style-type: none"> bypass reg 	bypass	TODO
IDIREG_SUB	Mux	<ul style="list-style-type: none"> bypass reg 	bypass	TODO
INREG_CTRL_AX	Mux	<ul style="list-style-type: none"> bypass reg 	bypass	TODO
INREG_CTRL_AY	Mux	<ul style="list-style-type: none"> bypass reg 	bypass	TODO

continues on next page

Table 3 – continued from previous page

Name	Type	Values	Default	Documentation
INREG_CTRL_AZ	Mux	<ul style="list-style-type: none"> • bypass • reg 	bypass	TODO
INREG_CTRL_BX	Mux	<ul style="list-style-type: none"> • bypass • reg 	bypass	TODO
INREG_CTRL_BY	Mux	<ul style="list-style-type: none"> • bypass • reg 	bypass	TODO
INREG_CTRL_BZ	Mux	<ul style="list-style-type: none"> • bypass • reg 	bypass	TODO
MODE	Mux	<ul style="list-style-type: none"> • three_9x9 • two_18x19 • one_27x27 • sum_of_2_18x19 • one_18x18_plus_36 	two_18x19	TODO
NCLR0_INV	Bool	t/f	f	TODO
NCLR0_SEL	Mux	<ul style="list-style-type: none"> • labclk3 • lsim2 	labclk3	TODO
NCLR1_INV	Bool	t/f	f	TODO
NCLR1_SEL	Mux	<ul style="list-style-type: none"> • labclk4 • lsim3 	labclk4	TODO
OREG_CTRL	Mux	<ul style="list-style-type: none"> • bypass • reg 	bypass	TODO
PARTIAL_RECONFIG_EN	Bool	t/f	f	TODO
PREADDER_EN	Mux	<ul style="list-style-type: none"> • off • add • sub 	off	TODO
PRELOAD	Ram	00-3f	0	TODO
PRELOAD_INV	Bool	t/f	f	TODO
PROGINV	Ram	108 bits	0	TODO
SUB_INV	Bool	t/f	f	TODO
SYSTOLIC_REG_EN	Bool	t/f	f	TODO

2.2.4 M10K

The M10K blocks provide 10240 (256*40) bits of dual-ported rom or ram.

TODO: everything, GOUT/GIN/DCMUX mapping is done

Name	Instance	Type	Values	Default	Documentation
A_ADDCLR_EN		Bool	t/f	f	TODO
A_DATA_FLOW_THRU		Bool	t/f	f	TODO
A_DATA_WIDTH		Num	<ul style="list-style-type: none"> • 1-2 • 5 • 10 • 20 • 40 	40	TODO
A_DMY_PWDWN		Ram	0-f	6	TODO
A_FAST_READ		Bool	t/f	f	TODO
A_FAST_WRITE		Mux	<ul style="list-style-type: none"> • off • fast • slow 	off	TODO
A_OUTCLR_EN		Mux	<ul style="list-style-type: none"> • off • reg • lat 	off	TODO
A_OUTEN_DELAY		Ram	0-7	1	TODO
A_OUTEN_PULSE		Ram	0-3	3	TODO
A_OUTPUT_SEL		Mux	<ul style="list-style-type: none"> • async • reg 	async	TODO
A_SAEN_DELAY		Ram	0-7	0	TODO
A_SA_WREN_DELAY		Ram	0-3	0	TODO
A_WL_DELAY		Ram	0-3	1	TODO
A_WR_TIMER_PULSE		Ram	00-1f	06	TODO
BIST_MODE		Bool	t/f	f	TODO
BOT_1_ADDCLR_SEL		Num	<ul style="list-style-type: none"> • 0-1 	0	TODO
BOT_1_CORECLK_SEL		Num	<ul style="list-style-type: none"> • 0-1 	0	TODO
BOT_1_INCLK_SEL		Num	<ul style="list-style-type: none"> • 0-1 	0	TODO
BOT_1_OUTCLK_SEL		Num	<ul style="list-style-type: none"> • 0-1 	0	TODO

continues on next page

Table 4 – continued from previous page

Name	Instance	Type	Values	Default	Documentation
BOT_1_OUTCLR_SEL		Num	• 0-1	0	TODO
BOT_CE0_INV		Bool	t/f	f	TODO
BOT_CE0_SEL		Num	• 0-1	0	TODO
BOT_CE1_INV		Bool	t/f	f	TODO
BOT_CE1_SEL		Num	• 0-1	0	TODO
BOT_CLK_INV		Bool	t/f	f	TODO
BOT_CLK_SEL		Num	• 0-1	0	TODO
BOT_CLR_INV		Bool	t/f	f	TODO
BOT_CLR_SEL		Num	• 0-1	0	TODO
BOT_CORECLK_SEL		Num	• 0-2	0	TODO
BOT_INCLK_SEL		Num	• 0-2	0	TODO
BOT_OUTCLK_SEL		Num	• 0-1	0	TODO
BOT_R_INV		Bool	t/f	f	TODO
BOT_R_SEL		Num	• 0-2	0	TODO
BOT_W_INV		Bool	t/f	f	TODO
BOT_W_SEL		Num	• 0-2	0	TODO
B_ADDCLR_EN		Bool	t/f	f	TODO
B_DATA_FLOW_THRU		Bool	t/f	f	TODO
B_DATA_WIDTH		Num	• 1-2 • 5 • 10 • 20 • 40	1	TODO
B_DMY_DELAY		Ram	0-3	1	TODO
B_DMY_DELAY		Ram	0-3	1	TODO
B_DMY_PWDWN		Ram	0-f	6	TODO
B_FAST_READ		Bool	t/f	f	TODO

continues on next page

Table 4 – continued from previous page

Name	Instance	Type	Values	Default	Documentation
B_FAST_WRITE		Mux	<ul style="list-style-type: none"> • off • fast • slow 	off	TODO
B_OUTCLR_EN		Mux	<ul style="list-style-type: none"> • off • reg • lat 	off	TODO
B_OUTEN_DELAY		Ram	0-7	1	TODO
B_OUTEN_PULSE		Ram	0-3	3	TODO
B_OUTPUT_SEL		Mux	<ul style="list-style-type: none"> • async • reg 	async	TODO
B_SAEN_DELAY		Ram	0-7	0	TODO
B_SA_WREN_DELAY		Ram	0-3	0	TODO
B_WL_DELAY		Ram	0-3	1	TODO
B_WR_TIMER_PULSE		Ram	00-1f	06	TODO
DISABLE_UNUSED		Bool	t/f	t	TODO
ITG_LFSR		Bool	t/f	f	TODO
PACK_MODE		Bool	t/f	f	TODO
PR_EN		Bool	t/f	f	TODO
TDF_ATPG		Bool	t/f	f	TODO
TEST_MODE_OFF		Bool	t/f	t	TODO
TOP_ADDCLR_SEL		Num	<ul style="list-style-type: none"> • 0-1 	0	TODO
TOP_CE0_INV		Bool	t/f	f	TODO
TOP_CE0_SEL		Num	<ul style="list-style-type: none"> • 0-1 	0	TODO
TOP_CE1_INV		Bool	t/f	f	TODO
TOP_CE1_SEL		Num	<ul style="list-style-type: none"> • 0-1 	0	TODO
TOP_CLK_INV		Bool	t/f	f	TODO
TOP_CLK_SEL		Num	<ul style="list-style-type: none"> • 0-1 	0	TODO
TOP_CLR_INV		Bool	t/f	f	TODO
TOP_CLR_SEL		Num	<ul style="list-style-type: none"> • 0-1 	0	TODO
TOP_CORECLK_SEL		Num	<ul style="list-style-type: none"> • 0-2 	0	TODO

continues on next page

Table 4 – continued from previous page

Name	Instance	Type	Values	Default	Documentation
TOP_INCLK_SEL		Num	• 0-2	0	TODO
TOP_OUTCLK_SEL		Num	• 0-1	0	TODO
TOP_OUTCLR_SEL		Num	• 0-1	0	TODO
TOP_R_INV		Bool	t/f	f	TODO
TOP_R_SEL		Num	• 0-2	0	TODO
TOP_W_INV		Bool	t/f	f	TODO
TOP_W_SEL		Num	• 0-2	0	TODO
TRUE_DUAL_PORT		Bool	t/f	f	TODO
RAM	0-255	Ram	40 bits	0	TODO

2.3 Peripheral logic blocks

2.3.1 GPIO

The GPIO blocks connect the FPGA with the exterior through the package pins. Each block controls 4 pads, which are connected to up to 4 pins.

TODO: everything, GOUT/GIN/DCMUX mapping is done

Name	Instance	Type	Values	Default	Documentation
IOCSR_STD	0-3	Mux	<ul style="list-style-type: none"> nvr_high nvr_low vr dis 	nvr_high	TODO
OUTPUT_DUTY_CYCLE_DELAY_HOLD	0-3	Bool	t/f	f	TODO
OUTPUT_DUTY_CYCLE_DELAY_PSUM	0-3	Sum	<ul style="list-style-type: none"> 0 50 100 150 	0	TODO
OUTPUT_DUTY_CYCLE_DELAY_RISE	0-3	Bool	t/f	f	TODO
PLL_SELECT	0-3	Mux	<ul style="list-style-type: none"> codin pll 	codin	TODO
SLEW_RATE_SLOW	0-3	Bool	t/f	f	TODO
TERMINATION_CONTROL	0-3	Mux	<ul style="list-style-type: none"> regio rupdn 	regio	TODO
TERMINATION_CONTROL_SHIFT	0-3	Bool	t/f	f	TODO
TERMINATION_MODE	0-3	Mux	<ul style="list-style-type: none"> pds rs_static rt_pds_dynamic rt_rs_dynamic rt_static 	pds	TODO
USE_BUS_HOLD	0-3	Bool	t/f	f	TODO
USE_OPEN_DRAIN	0-3	Bool	t/f	f	TODO
USE_PCI_DIODE_CLAMP	0-3	Bool	t/f	f	TODO
USE_WEAK_PULLUP	0-3	Bool	t/f		TODO
DRIVE_STRENGTH	0-3	Mux	<ul style="list-style-type: none"> off prog_gnd prog_pwr lvds_1r lvds_3r v3p0_pci_pcix v3p0_lvttl_4ma v3p0_lvttl_8ma v3p0_lvttl_12ma v3p0_lvttl_16ma v3p3_lvttl_4ma 		TODO
20			<ul style="list-style-type: none"> v3p0_lvcmos_4ma v3p0_lvcmos_8ma v3p0_lvcmos_12ma 		Chapter 2. CycloneV internals description

2.3.2 DQS16

The DQS16 blocks handle differential signaling protocols. Each supervises 4 GPIO blocks for a total of 16 signals, hence their name.

TODO: everything

Name	Instance	Type	Values	Default	Documentation
ADDR_DQS_DELAY_CHAIN_LENGTH	ADDR_DQS_DELAY_CHAIN_LENGTH	Ctrl	0-3	0	TODO
DELAY_CHAIN_CONTROL_INPUT	DELAY_CHAIN_CONTROL_INPUT	Mux	<ul style="list-style-type: none"> • dll1in • dll2in • core_in • sel_0 	dll1in	TODO
DELAY_CHAIN_LATCHES_BYPASS	DELAY_CHAIN_LATCHES_BYPASS	Bool	t/f	f	TODO
DFT_RB_RSCAN_OVRD_REG_EN	DFT_RB_RSCAN_OVRD_REG_EN	Bool	t/f	f	TODO
DFT_RB_RSCAN_OVRD_TDF_EN	DFT_RB_RSCAN_OVRD_TDF_EN	Bool	t/f	f	TODO
DQS_BUS_WIDTH	DQS_BUS_WIDTH	Num	<ul style="list-style-type: none"> • 0 • 8 • 16 • 32 	8	TODO
DQS_DELAY_CHAIN_PWDOWN_DFT_OVRD_DIS	DQS_DELAY_CHAIN_PWDOWN_DFT_OVRD_DIS	Bool	t/f	t	TODO
DQS_DELAY_CHAIN_PWDOWN_DQS_OVRD_DIS	DQS_DELAY_CHAIN_PWDOWN_DQS_OVRD_DIS	Bool	t/f	f	TODO
DQS_DELAY_CHAIN_RB_ADDI_EN	DQS_DELAY_CHAIN_RB_ADDI_EN	Bool	t/f	f	TODO
DQS_DELAY_CHAIN_RB_CO	DQS_DELAY_CHAIN_RB_CO	Ram	0-3	3	TODO
DQS_DELAY_CHAIN_TWO_DLY_EN	DQS_DELAY_CHAIN_TWO_DLY_EN	Bool	t/f	t	TODO
DQS_ENABLE_SEL	DQS_ENABLE_SEL	Mux	<ul style="list-style-type: none"> • combi_pst • pst • ht_pst • pst_ena 	combi_pst	TODO
DQS_PHASE_TRANSFER_NEG_EN	DQS_PHASE_TRANSFER_NEG_EN	Bool	t/f	f	TODO
DQS_POSTAMBLE_EN	DQS_POSTAMBLE_EN	Bool	t/f	f	TODO
DQS_POSTAMBLE_NEJ_SEL	DQS_POSTAMBLE_NEJ_SEL	Mux	<ul style="list-style-type: none"> • cff • ip_sc 	cff	TODO
DQS_PWR_SVG_EN	DQS_PWR_SVG_EN	Bool	t/f	t	TODO
HR_CLK_PST_INV	HR_CLK_PST_INV	Bool	t/f	t	TODO
HR_CLK_PST_SEL	HR_CLK_PST_SEL	Mux	<ul style="list-style-type: none"> • dqs_clkout • seq_hr_clk 	seq_hr_clk	TODO
PST_DQS_CLK_INV_PHASE_INV	PST_DQS_CLK_INV_PHASE_INV	Bool	t/f	f	TODO

continues on next page

Table 5 – continued from previous page

Name	Instance	Type	Values	Default	Documentation
PST_DQS_CLK_INV_PHASE_SEL		Mux	<ul style="list-style-type: none"> cff ip_sc 	cff	TODO
PST_DQS_DELAY_CHAIN_LENGTH		Ram	0-3	0	TODO
PST_USE_PHASECTRLIN		Bool	t/f	f	TODO
RBT_BYPASS_VAL		Ram	0-1	0	TODO
RBT_NEJ_OCT_HALFT_EN		Bool	t/f	f	TODO
RB_2X_CLK_DQS_EN		Bool	t/f	f	TODO
RB_2X_CLK_DQS_INV		Bool	t/f	f	TODO
RB_2X_CLK_OCT_EN		Bool	t/f	f	TODO
RB_2X_CLK_OCT_INV		Bool	t/f	f	TODO
RB_ACLR_LFIFO_EN		Bool	t/f	f	TODO
RB_ACLR_PST_EN		Bool	t/f	f	TODO
RB_BYP_OCT_SEL		Mux	<ul style="list-style-type: none"> combi reg reg_2x bypass_val 	bypass_val	TODO
RB_CLK_AC_EN		Bool	t/f	f	TODO
RB_CLK_AC_INV		Bool	t/f	t	TODO
RB_CLK_DQ_EN		Bool	t/f	f	TODO
RB_CLK_HR_EN		Bool	t/f	f	TODO
RB_CLK_OP_EN		Bool	t/f	f	TODO
RB_CLK_OP_SEL		Mux	<ul style="list-style-type: none"> clk0 delay_clk 	clk0	TODO
RB_CLK_PST_EN		Bool	t/f	f	TODO
RB_FIFO_WEN_EN		Bool	t/f	f	TODO
RB_FR_CLK_OCT_EN		Bool	t/f	f	TODO
RB_FR_CLK_OCT_INV		Bool	t/f	f	TODO
RB_FR_CLK_OCT_SEL		Mux	<ul style="list-style-type: none"> clk_out_1 seq_hr_clk 	clk_out_1	TODO
RB_HR_BYPASS_CFF_EN		Bool	t/f	t	TODO
RB_HR_BYPASS_SEL_IPEN		Mux	<ul style="list-style-type: none"> cff ip_sc 	cff	TODO
RB_HR_CLK_OCT_EN		Bool	t/f	f	TODO
RB_HR_CLK_OCT_INV		Bool	t/f	f	TODO

continues on next page

Table 5 – continued from previous page

Name	Instance	Type	Values	Default	Documentation
RB_HR_CLK_OCT_SEL		Mux	<ul style="list-style-type: none"> clk_out_1 seq_hr_clk 	clk_out_1	TODO
RB_LFIFO		Ram	32 bits	0	TODO
RB_LFIFO_BYPASS		Bool	t/f	t	TODO
RB_LFIFO_OCT_EN		Bool	t/f	t	TODO
RB_LFIFO_PHY_CLK_INV		Bool	t/f	f	TODO
RB_LFIFO_PHY_CLK_SEL		Ram	0-1	0	TODO
RB_T11_GATING_SEL_CFF		Ram	00-1f	0	TODO
RB_T11_GATING_SEL_IPEN		Mux	<ul style="list-style-type: none"> cff ip_sc 	cff	TODO
RB_T11_UNGATING_SEL_CFF		Ram	00-1f	0	TODO
RB_T11_UNGATING_SEL_IPEN		Mux	<ul style="list-style-type: none"> cff ip_sc 	cff	TODO
RB_T7_DQS_SEL_DQS_IPEN		Mux	<ul style="list-style-type: none"> cff ip_sc 	cff	TODO
RB_T7_SEL_IREG_CFF_DELAY		Ram	00-1f	0	TODO
RB_T9_SEL_OCT_CFF		Ram	00-1f	0	TODO
RB_T9_SEL_OCT_IPEN		Mux	<ul style="list-style-type: none"> cff ip_sc 	cff	TODO
RB_VFIFO_EN		Bool	t/f	f	TODO
RDFT_ITG_XOR_EN		Bool	t/f	f	TODO
RXCLK_01_SEL		Ram	0-1	0	TODO
RXCLK_45_SEL		Ram	0-1	0	TODO
RXCLK_89_SEL		Ram	0-1	0	TODO
RXCLK_CD_SEL		Ram	0-1	0	TODO
TXCLK_23_SEL		Ram	0-1	0	TODO
TXCLK_67_SEL		Ram	0-1	0	TODO
TXCLK_AB_SEL		Ram	0-1	0	TODO
TXCLK_EF_SEL		Ram	0-1	0	TODO
UPDATE_ENABLE_INPUT		Mux	<ul style="list-style-type: none"> sel1 sel2 core sel0 	sel1	TODO
BITSLLIP_CFG	0-15	Num	<ul style="list-style-type: none"> 1-11 	1	TODO
CE_OEREG_TIEOFF_EN		Bool	t/f	f	TODO

continues on next page

Table 5 – continued from previous page

Name	Instance	Type	Values	Default	Documentation
CE_OUTREG_TIEOFF_EN	0-15	Bool	t/f	f	TODO
DDIO_OE_EN	0-15	Bool	t/f	f	TODO
DQS_CLK_SEL	0-15	Mux	<ul style="list-style-type: none"> • clkout0 • dq_clk • dqs_clk • addr_clk 	clkout0	TODO
FIFO_MODE_SEL	0-15	Mux	<ul style="list-style-type: none"> • fifo_hr_mode • fifo_fr_mode • bitsslip_mode • des_bs_input • des_io_input • ser_output 	fifo_hr_mode	TODO
FIFO_RCLK_IPEN	0-15	Mux	<ul style="list-style-type: none"> • cff • ip_sc 	cff	TODO
FIFO_RCLK_SEL	0-15	Mux	<ul style="list-style-type: none"> • clkln1 • dqs_clk • seq_hr_clk • vcc 	vcc	TODO
INPUT_PATH_CE_ON	0-15	Bool	t/f	f	TODO
INPUT_REG0_SEL	0-15	Mux	<ul style="list-style-type: none"> • sel_bypass • sel_group_fifo0 • sel_cdatamxin0 • sel_cdatamxin5 	sel_bypass	TODO

continues on next page

Table 5 – continued from previous page

Name	Instance	Type	Values	Default	Documentation
INPUT_REG1_SEL	IO-15	Mux	<ul style="list-style-type: none"> • sel_bypass • sel_group_fifo1 • sel_cdatamxin1 • sel_cdatamxin6 	sel_bypass	TODO
INPUT_REG2_SEL	IO-15	Mux	<ul style="list-style-type: none"> • sel_bypass • sel_group_fifo2 • sel_cdatamxin2 • sel_cdatamxin7 	sel_bypass	TODO
INPUT_REG3_SEL	IO-15	Mux	<ul style="list-style-type: none"> • sel_bypass • sel_group_fifo3 • sel_cdatamxin3 • sel_cdatamxin8 	sel_bypass	TODO
INPUT_REG4_SEL	IO-15	Mux	<ul style="list-style-type: none"> • sel_bypass • sel_locked_dpa • sel_cdatamxin4 • sel_cdatamxin9 	sel_bypass	TODO
INREG_POWER_UP_STATE	IO-15	Ram	0-1	0	TODO
INREG_SCLR_EN	IO-15	Bool	t/f	f	TODO
INREG_SCLR_VAL	IO-15	Ram	0-1	0	TODO
IOREG_PWR_SVC_EN	IO-15	Bool	t/f	t	TODO
IP_SC_OR_FIFO_SEL	IO-15	Mux	<ul style="list-style-type: none"> • cff • ip_sc 	cff	TODO
IR_FIFO_RCLK_EN	IO-15	Bool	t/f	f	TODO
IR_FIFO_TCLK_EN	IO-15	Bool	t/f	f	TODO
OEREG_ACLR_EN	IO-15	Bool	t/f	f	TODO

continues on next page

Table 5 – continued from previous page

Name	Instance	Type	Values	Default	Documentation
OEREG_CLK_INV	0-15	Bool	t/f	f	TODO
OEREG_HR_CLK_EN	0-15	Bool	t/f	f	TODO
OEREG_OUTPUT_SEL	0-15	Mux	<ul style="list-style-type: none"> • sel_oe0 • sel_1x • sel_1x_delay • sel_2x 	sel_oe0	TODO
OEREG_POWER_UP_STATE	0-1	Ram	0-1	0	TODO
OEREG_SCLR_DELAY	0-15	Ram	0-1	0	TODO
OEREG_SCLR_EN	0-15	Bool	t/f	f	TODO
OE_2X_CLK_EN	0-15	Bool	t/f	f	TODO
OE_2X_CLK_INV	0-15	Bool	t/f	f	TODO
OE_HALF_RATE_BYPASS	0-1	Bool	t/f	t	TODO
OE_HALF_RATE_OPEN	0-15	Mux	<ul style="list-style-type: none"> • cff • ip_sc 	cff	TODO
OUTREG_MODE_SEL	0-1	Mux	<ul style="list-style-type: none"> • sdr • ddr 	sdr	TODO
OUTREG_OUTPUT_SEL	0-15	Mux	<ul style="list-style-type: none"> • sel_iodout0 • sel_sdr • sel_sdr_delay • sel_2xff 	sel_iodout0	TODO
OUTREG_POWER_UP_STATE	0-1	Ram	0-1	0	TODO
OUTREG_SCLR_EN	0-15	Bool	t/f	f	TODO
OUTREG_SCLR_VAL	0-15	Ram	0-1	0	TODO
RBE_HRATE_CLK_SEL	0-15	Mux	<ul style="list-style-type: none"> • clkout1 • hr_clk 	clkout1	TODO
RBOE_LVL_FR_CLK_EN	0-1	Bool	t/f	f	TODO
RBOE_LVL_FR_CLK_INV	0-1	Bool	t/f	f	TODO
RB_FIFO_WCLK_EN	0-15	Bool	t/f	f	TODO
RB_FIFO_WCLK_INV	0-15	Bool	t/f	f	TODO
RB_FIFO_WCLK_SEL	0-15	Mux	<ul style="list-style-type: none"> • clkin0 • dqs_bus 	clkin0	TODO
RB_IREG_T1T1_BYPASS_EN	0-1	Bool	t/f	f	TODO
RB_OEO_INV	0-15	Bool	t/f	t	TODO
RB_T1_SEL_IREG_CFF_DELAY	0-15	Ram	00-1f	0	TODO

continues on next page

Table 5 – continued from previous page

Name	Instance	Type	Values	Default	Documentation
RB_T1_SEL_IREF	C0-IPEN	Mux	<ul style="list-style-type: none"> cff ip_sc 	cff	TODO
RB_T9_SEL_EREC	C0-15FF_DELAY	Ram	00-1f	0	TODO
RB_T9_SEL_EREC	C0-IPEN	Mux	<ul style="list-style-type: none"> cff ip_sc 	cff	TODO
RB_T9_SEL_OREC	C0-15FF_DELAY	Ram	00-1f	0	TODO
RB_T9_SEL_OREC	C0-IPEN	Mux	<ul style="list-style-type: none"> cff ip_sc 	cff	TODO
SET_T3_FOR_CDAT	A150IN	Ram	0-7	0	TODO
SET_T3_FOR_CDAT	A151IN	Ram	0-7	0	TODO
TXOUT_FCLK_SEL	L15	Mux	<ul style="list-style-type: none"> txout fclk 	txout	TODO
USE_CLR_INREG	G0EN	Bool	t/f	f	TODO
USE_CLR_OUTREG	G1EN	Bool	t/f	f	TODO

2.3.3 FPLL

The Fractional PLL blocks synthesize 9 frequencies from an input with integer or fractional ratios.

TODO: everything, GOUT/GIN/DCMUX mapping is done

Name	Instance	Type	Values	Default	Documentation
ATB		Ram	0-f	0	TODO
AUTO_CLK_SW_EN		Bool	t/f	f	TODO
BWCTRL		Ram	0-f	4	TODO
C0_COUT_EN		Bool	t/f	f	TODO
C0_EXTCLK_DLLOUT_EN		Bool	t/f	f	TODO
C1_COUT_EN		Bool	t/f	f	TODO
C1_EXTCLK_DLLOUT_EN		Bool	t/f	f	TODO
C2_COUT_EN		Bool	t/f	f	TODO
C2_EXTCLK_DLLOUT_EN		Bool	t/f	f	TODO
C3_COUT_EN		Bool	t/f	f	TODO
C3_EXTCLK_DLLOUT_EN		Bool	t/f	f	TODO
C4_COUT_EN		Bool	t/f	f	TODO
C5_COUT_EN		Bool	t/f	f	TODO
C6_COUT_EN		Bool	t/f	f	TODO
C7_COUT_EN		Bool	t/f	f	TODO
C8_COUT_EN		Bool	t/f	f	TODO
CLKIN_0_SRC		Ram	0-f	2	TODO
CLKIN_1_SRC		Ram	0-f	3	TODO
CLK_LOSS_EDGE		Ram	0-1	0	TODO

continues on next page

Table 6 – continued from previous page

Name	Instance	Type	Values	Default	Documentation
CLK_LOSS_SW_EN		Bool	t/f	f	TODO
CLK_SW_DELAY		Ram	0-7	0	TODO
CMP_BUF_DELAY		Ram	0-7	0	TODO
CP_COMP		Bool	t/f	f	TODO
CP_CURRENT		Ram	0-7	2	TODO
CTRL_OVERRIDE_SETTING		Bool	t/f	t	TODO
DLL_SRC		Ram	00-1f	1c	TODO
DPADIV_VCOPH_DIV		Ram	0-3	0	TODO
DPRIO0_BASE_ADDR		Ram	00-3f	0	TODO
DPRIO_DPS_ATPGMODE_INVERT		Bool	t/f	f	TODO
DPRIO_DPS_CLK_INVERT		Bool	t/f	f	TODO
DPRIO_DPS_CSR_TEST_INVERT		Bool	t/f	f	TODO
DPRIO_DPS_ECN_MUX		Ram	0-1	0	TODO
DPRIO_DPS_RESERVED_INVERT		Bool	t/f	f	TODO
DPRIO_DPS_RST_N_INVERT		Bool	t/f	f	TODO
DPRIO_DPS_SCANEN_INVERT		Bool	t/f	f	TODO
DSM_DITHER		Ram	0-3	0	TODO
DSM_OUT_SEL		Ram	0-3	0	TODO
DSM_RESET		Bool	t/f	f	TODO
ECN_BYPASS		Bool	t/f	f	TODO
ECN_TEST_EN		Bool	t/f	f	TODO
FBCLK_MUX_1		Ram	0-3	0	TODO
FBCLK_MUX_2		Ram	0-1	0	TODO
FORCELOCK		Bool	t/f	f	TODO
FPLL_ENABLE		Bool	t/f	f	TODO
FRACTIONAL_CARRY_OUT		Ram	0-3	3	TODO
FRACTIONAL_DIVISION_SETTING		Ram	32 bits	0	TODO
FRACTIONAL_VALUE_READY		Bool	t/f	t	TODO
LF_TESTEN		Bool	t/f	f	TODO
LOCK_FILTER_CFG_SETTING		Ram	000-fff	001	TODO
LOCK_FILTER_TEST		Bool	t/f	f	TODO
MANUAL_CLK_SW_EN		Bool	t/f	f	TODO
M_CNT_BYPASS_EN		Bool	t/f	f	TODO
M_CNT_COARSE_DELAY		Ram	0-7	0	TODO
M_CNT_FINE_DELAY		Ram	0-3	0	TODO
M_CNT_HI_DIV_SETTING		Ram	00-ff	01	TODO
M_CNT_IN_SRC		Ram	0-3	0	TODO
M_CNT_LO_DIV_SETTING		Ram	00-ff	01	TODO
M_CNT_LO_PRESET_SETTING		Ram	00-ff	01	TODO
M_CNT_ODD_DIV_DUTY_EN		Bool	t/f	f	TODO
M_CNT_PH_MUX_PRESET_SETTING		Ram	0-7	0	TODO
NREVERT_INVERT		Bool	t/f	f	TODO
N_CNT_BYPASS_EN		Bool	t/f	f	TODO
N_CNT_COARSE_DELAY		Ram	0-7	0	TODO
N_CNT_FINE_DELAY		Ram	0-3	0	TODO
N_CNT_HI_DIV_SETTING		Ram	00-ff	01	TODO
N_CNT_LO_DIV_SETTING		Ram	00-ff	01	TODO
N_CNT_ODD_DIV_DUTY_EN		Bool	t/f	f	TODO
PL_AUX_ATB		Bool	t/f	f	TODO

continues on next page

Table 6 – continued from previous page

Name	Instance	Type	Values	Default	Documentation
PL_AUX_ATB_COMP_MINUS		Bool	t/f	f	TODO
PL_AUX_ATB_COMP_PLUS		Bool	t/f	f	TODO
PL_AUX_ATB_EN0		Bool	t/f	f	TODO
PL_AUX_ATB_EN0_PRECOMP		Bool	t/f	f	TODO
PL_AUX_ATB_EN1		Bool	t/f	f	TODO
PL_AUX_ATB_EN1_PRECOMP		Bool	t/f	f	TODO
PL_AUX_ATB_MODE		Ram	00-1f	0	TODO
PL_AUX_BG_KICKSTART		Bool	t/f	f	TODO
PL_AUX_BG_POWERDOWN		Bool	t/f	f	TODO
PL_AUX_BYPASS_MODE_CTRL_CURRENT		Bool	t/f	f	TODO
PL_AUX_BYPASS_MODE_CTRL_VOLTAGE		Bool	t/f	f	TODO
PL_AUX_COMP_POWERDOWN		Bool	t/f	f	TODO
PL_AUX_VBGMON_POWERDOWN		Bool	t/f	f	TODO
PM_AUX_CAL_CLK_TEST_SEL		Bool	t/f	f	TODO
PM_AUX_CAL_RESULT_STATUS		Bool	t/f	f	TODO
PM_AUX_IQCLK_CAL_CLK_SEL		Ram	0-7	0	TODO
PM_AUX_RX_IMP		Ram	0-3	0	TODO
PM_AUX_TERM_CAL		Bool	t/f	f	TODO
PM_AUX_TERM_CAL_RX_OVER_VAL		Ram	00-1f	0	TODO
PM_AUX_TERM_CAL_RX_OVER_VAL_EN		Bool	t/f	f	TODO
PM_AUX_TERM_CAL_TX_OVER_VAL		Ram	00-1f	0	TODO
PM_AUX_TERM_CAL_TX_OVER_VAL_EN		Bool	t/f	f	TODO
PM_AUX_TEST_COUNTER		Bool	t/f	f	TODO
PM_AUX_TX_IMP		Ram	0-3	0	TODO
REF_BUF_DELAY		Ram	0-7	0	TODO
REGULATION_BYPASS		Bool	t/f	f	TODO
REG_BOOST		Ram	0-7	0	TODO
RIPPLECAP_CTRL		Ram	0-3	0	TODO
SLF_RST		Ram	0-3	0	TODO
SW_REFCLK_SRC		Ram	0-1	0	TODO
TCLK_MUX_EN		Bool	t/f	f	TODO
TCLK_SEL		Ram	0-1	1	TODO
TESTDN_ENABLE		Bool	t/f	f	TODO
TESTUP_ENABLE		Bool	t/f	f	TODO
TEST_ENABLE		Bool	t/f	f	TODO
UNLOCK_FILTER_CFG_SETTING		Ram	0-7	0	TODO
VC0DIV_OVERRIDE		Bool	t/f	t	TODO
VCCD0G_ATB		Ram	0-3	0	TODO
VCCD0G_OUTPUT		Ram	0-7	0	TODO
VCCD1G_ATB		Ram	0-3	0	TODO
VCCD1G_OUTPUT		Ram	0-7	0	TODO
VCCM1G_TAP		Ram	0-f	b	TODO
VCCR_PD		Bool	t/f	f	TODO
VCO0PH_EN		Bool	t/f	f	TODO
VCO_DIV		Ram	0-1	1	TODO
VCO_PH0_EN		Bool	t/f	f	TODO
VCO_PH1_EN		Bool	t/f	f	TODO
VCO_PH2_EN		Bool	t/f	f	TODO
VCO_PH3_EN		Bool	t/f	f	TODO

continues on next page

Table 6 – continued from previous page

Name	Instance	Type	Values	Default	Documentation
VCO_PH4_EN		Bool	t/f	f	TODO
VCO_PH5_EN		Bool	t/f	f	TODO
VCO_PH6_EN		Bool	t/f	f	TODO
VCO_PH7_EN		Bool	t/f	f	TODO
VCTRL_TEST_VOLTAGE		Ram	0-7	3	TODO
EXTCLK_CNT_SRC	0-1	Ram	00-1f	1c	TODO
EXTCLK_ENABLE	0-1	Bool	t/f	t	TODO
EXTCLK_INVERT	0-1	Bool	t/f	f	TODO
BYPASS_EN	0-8	Bool	t/f	f	TODO
CNT_COARSE_DELAY	0-8	Ram	0-7	0	TODO
CNT_FINE_DELAY	0-8	Ram	0-3	0	TODO
CNT_IN_SRC	0-8	Ram	0-3	2	TODO
CNT_PH_MUX_PRESET	0-8	Ram	0-7	0	TODO
CNT_PRESET	0-8	Ram	00-ff	01	TODO
DPRIO0_CNT_HI_DIV	0-8	Ram	00-ff	01	TODO
DPRIO0_CNT_LO_DIV	0-8	Ram	00-ff	01	TODO
DPRIO0_CNT_ODD_DIV_EVEN_DUTY_EN	0-8	Bool	t/f	f	TODO
SRC	0-8	Bool	t/f	f	TODO
LOADEN_COARSE_DELAY	0-1	Ram	0-7	0	TODO
LOADEN_ENABLE	0-1	Bool	t/f	f	TODO
LOADEN_FINE_DELAY	0-1	Ram	0-3	0	TODO
LVDSCLK_COARSE_DELAY	0-1	Ram	0-7	0	TODO
LVDSCLK_ENABLE	0-1	Bool	t/f	f	TODO
LVDSCLK_FINE_DELAY	0-1	Ram	0-3	0	TODO

2.3.4 CBUF

Name	Instance	Type	Values	Default	Documentation
EFB_MUX		Ram	0-1	0	TODO
EFB_MUX_EN		Bool	t/f	f	TODO
EXTCLKOUT_MUX_EN		Bool	t/f	f	TODO
FBIN_MUX	0-1	Ram	0-1	0	TODO
MUX0	0-1	Ram	0-1	0	TODO
MUX0_EN	0-1	Bool	t/f	f	TODO
MUX1	0-1	Ram	0-1	0	TODO
MUX1_EN	0-1	Bool	t/f	f	TODO
MUX2	0-1	Ram	0-1	0	TODO
MUX2_EN	0-1	Bool	t/f	f	TODO
MUX3	0-1	Ram	0-1	0	TODO
MUX3_EN	0-1	Bool	t/f	f	TODO
VCOPH_MUX	0-1	Ram	0-1	0	TODO
VCOPH_MUX_EN	0-1	Bool	t/f	f	TODO

2.3.5 CMUXC

The three or four Corner CMUX drives 3 horizontal RCLK grids and 3 vertical each.

Name	Instance	Type	Values	Default	Documentation
CLKPIN_INPUT_SELECT_0	SELECT_0	Mux	<ul style="list-style-type: none"> pin0 pin2 	pin0	TODO
CLKPIN_INPUT_SELECT_1	SELECT_1	Mux	<ul style="list-style-type: none"> pin1 pin3 	pin1	TODO
ENABLE_REGISTER_MODE	EN_MODE	Mux	<ul style="list-style-type: none"> enout reg1_enout reg2_enout vcc 	vcc	TODO
ENABLE_REGISTER_POWER_UP	EN_POWER_UP	Num	<ul style="list-style-type: none"> 0-1 	1	TODO
INPUT_SELECT	0-5	Ram	0-f	f	TODO
NCLKPIN_INPUT_SELECT_0	SELECT_0	Mux	<ul style="list-style-type: none"> npin0 npin2 	npin0	TODO
NCLKPIN_INPUT_SELECT_1	SELECT_1	Mux	<ul style="list-style-type: none"> npin1 npin3 	npin1	TODO
PLL_FEEDBACK_ENABLE_0		Mux	<ul style="list-style-type: none"> vcc pll_mcnt0 	vcc	TODO
PLL_FEEDBACK_ENABLE_1		Mux	<ul style="list-style-type: none"> vcc pll_mcnt0 	vcc	TODO
TOP_PRE_INPUT_SELECT_0	SELECT_0	Ram	00-1f	1f	TODO
TOP_PRE_INPUT_SELECT_1	SELECT_1	Ram	00-1f	1f	TODO
TOP_PRE_INPUT_SELECT_2	SELECT_2	Ram	00-1f	1f	TODO
TOP_PRE_INPUT_SELECT_3	SELECT_3	Ram	00-1f	1f	TODO

2.3.6 CMUXHG

The two Global Horizontal CMUX drive four GCLK grids each.

Name	Instance	Type	Values	Default	Documentation
BURST_COUNT	0-3	Ram	0-7	0	TODO

continues on next page

Table 7 – continued from previous page

Name	Instance	Type	Values	Default	Documentation
BURST_COUNT	CTRL	Mux	<ul style="list-style-type: none"> static core_ctrl 	static	TODO
BURST_EN	0-3	Bool	t/f	f	TODO
CLKPIN_INPUT	SELECT_0	Mux	<ul style="list-style-type: none"> pina pinb 	pina	TODO
CLKPIN_INPUT	SELECT_1	Mux	<ul style="list-style-type: none"> pina pinb 	pina	TODO
CLKPIN_INPUT	SELECT_2	Mux	<ul style="list-style-type: none"> pina pinb 	pina	TODO
CLKPIN_INPUT	SELECT_3	Mux	<ul style="list-style-type: none"> pina pinb 	pina	TODO
CLK_SELECT_A	0-3	Ram	0-3	0	TODO
CLK_SELECT_B	0-3	Ram	0-3	0	TODO
CLK_SELECT_C	0-3	Ram	0-3	0	TODO
CLK_SELECT_D	0-3	Ram	0-3	0	TODO
ENABLE_REGISTER	CTRL_MODE	Mux	<ul style="list-style-type: none"> enout reg1_enout reg2_enout vcc 	vcc	TODO
ENABLE_REGISTER	CTRL_POWER_UP	Num	<ul style="list-style-type: none"> 0-1 	1	TODO
INPUT_SELECT	0-3	Ram	00-3f	23	TODO
NCLKPIN_INPUT	SELECT_0	Mux	<ul style="list-style-type: none"> npina npinb 	npina	TODO
NCLKPIN_INPUT	SELECT_1	Mux	<ul style="list-style-type: none"> npina npinb 	npina	TODO
NCLKPIN_INPUT	SELECT_2	Mux	<ul style="list-style-type: none"> npina npinb 	npina	TODO

continues on next page

Table 7 – continued from previous page

Name	Instance	Type	Values	Default	Documentation
NCLKPIN_INPUT_SELECT_3	0-3	Mux	<ul style="list-style-type: none"> npina npinb 	npina	TODO
ORPHAN_PLL_INPUT_SELECT_0	0-3	Mux	<ul style="list-style-type: none"> orphan_pll0 orphan_pll3 	orphan_pll0	TODO
ORPHAN_PLL_INPUT_SELECT_1	0-3	Mux	<ul style="list-style-type: none"> orphan_pll1 orphan_pll4 	orphan_pll1	TODO
ORPHAN_PLL_INPUT_SELECT_2	0-3	Mux	<ul style="list-style-type: none"> orphan_pll2 orphan_pll5 	orphan_pll2	TODO
TESTSYN_ENOUT_SELECT	0-3	Mux	<ul style="list-style-type: none"> core_en pre_synenb 	core_en	TODO
DYNAMIC_CLK_SELECT		Bool	t/f	f	TODO
FEEDBACK_DRIVER_SELECT_0		Mux	<ul style="list-style-type: none"> in0_vcc in1 in2_vcc in3_vcc in4_vcc in5 in6 in7 	in0_vcc	TODO
FEEDBACK_DRIVER_SELECT_1		Mux	<ul style="list-style-type: none"> in0_vcc in1 in2_vcc in3_vcc in4_vcc in5 in6 in7 	in0_vcc	TODO
ORPHAN_PLL_FEEDBACK_OUT_SELECT_0	0-1	0-1	0	0	TODO
ORPHAN_PLL_FEEDBACK_OUT_SELECT_1	0-1	0-1	0	0	TODO

continues on next page

Table 7 – continued from previous page

Name	Instance	Type	Values	Default	Documentation
PLL_FEEDBACK_ENABLE_0		Mux	<ul style="list-style-type: none">• vcc• pll_mcnt0	vcc	TODO
PLL_FEEDBACK_ENABLE_1		Mux	<ul style="list-style-type: none">• vcc• pll_mcnt0	vcc	TODO
PLL_FEEDBACK_OUT_SELECT_0		Ram	0-1	0	TODO
PLL_FEEDBACK_OUT_SELECT_1		Ram	0-1	0	TODO

2.3.7 CMUXVG

The two Global Vertical CMUX drive four GCLK grids each.

Name	Instance	Type	Values	Default	Documentation
BURST_COUNT	0-3	Ram	0-7	0	TODO
BURST_COUNT_CTRL	CTRL	Mux	<ul style="list-style-type: none"> static core_ctrl 	static	TODO
BURST_EN	0-3	Bool	t/f	f	TODO
CLK_SELECT_A	0-3	Ram	0-3	0	TODO
CLK_SELECT_B	0-3	Ram	0-3	0	TODO
CLK_SELECT_C	0-3	Ram	0-3	0	TODO
CLK_SELECT_D	0-3	Ram	0-3	0	TODO
ENABLE_REGISTER_MODE	MODE	Mux	<ul style="list-style-type: none"> enout reg1_enout reg2_enout vcc 	vcc	TODO
ENABLE_REGISTER_POWER_UP	POWER_UP	Num	<ul style="list-style-type: none"> 0-1 	1	TODO
INPUT_SELECT	0-3	Ram	00-1f	1b	TODO
TESTSYN_ENOUT_SELECT	SELECT	Mux	<ul style="list-style-type: none"> core_en pre_synenb 	pre_synenb	TODO
DYNAMIC_CLK_SELECT		Bool	t/f	f	TODO
PLL_FEEDBACK_ENABLE_0		Mux	<ul style="list-style-type: none"> vcc pll_mcnt0 	vcc	TODO
PLL_FEEDBACK_ENABLE_1		Mux	<ul style="list-style-type: none"> vcc pll_mcnt0 	vcc	TODO
PLL_FEEDBACK_ENABLE_1		Mux	<ul style="list-style-type: none"> vcc pll_mcnt0 	vcc	TODO
PLL_FEEDBACK_ENABLE_2		Mux	<ul style="list-style-type: none"> vcc pll_mcnt0 	vcc	TODO
PLL_FEEDBACK_ENABLE_3		Mux	<ul style="list-style-type: none"> vcc pll_mcnt0 	vcc	TODO

2.3.8 CMUXHR

The two Regional Horizontal CMUX drive 12 vertical RCLK grids each, half on each side. Six are lost when touching the HPS.

Name	Instance	Type	Values	Default	Documentation
CLKPIN_INPUT_SELECT		Mux	<ul style="list-style-type: none"> pina pinb 	pina	TODO
ENABLE_REGISTER1MODE		Mux	<ul style="list-style-type: none"> enout reg1_enout reg2_enout vcc 	vcc	TODO
ENABLE_REGISTER1POWER_UP		Num	<ul style="list-style-type: none"> 0-1 	1	TODO
INPUT_SELECT	0-11	Ram	00-1f	13	TODO
NCLKPIN_INPUT_SELECT		Mux	<ul style="list-style-type: none"> npina npinb 	npina	TODO
BOT_PRE_INPUT_SELECT_0		Ram	00-1f	1f	TODO
BOT_PRE_INPUT_SELECT_1		Ram	00-1f	1f	TODO
BOT_PRE_INPUT_SELECT_2		Ram	00-1f	1f	TODO
BOT_PRE_INPUT_SELECT_3		Ram	00-1f	1f	TODO
FEEDBACK_DRIVER_SELECT_0		Mux	<ul style="list-style-type: none"> vcc orphan_pll_mcnto0 orphan_pll_mcnto1 orphan_pll_mcnto2 	vcc	TODO
FEEDBACK_DRIVER_SELECT_1		Mux	<ul style="list-style-type: none"> vcc orphan_pll_mcnto0 orphan_pll_mcnto1 orphan_pll_mcnto2 	vcc	TODO
PLL_FEEDBACK_ENABLE_0		Mux	<ul style="list-style-type: none"> vcc pll_mcnt0 	vcc	TODO
PLL_FEEDBACK_ENABLE_1		Mux	<ul style="list-style-type: none"> vcc pll_mcnt0 	vcc	TODO
PRE_INPUT_SELECT_0		Ram	00-1f	1f	TODO
PRE_INPUT_SELECT_1		Ram	00-1f	1f	TODO
PRE_INPUT_SELECT_2		Ram	00-1f	1f	TODO
PRE_INPUT_SELECT_3		Ram	00-1f	1f	TODO
TOP_PRE_INPUT_SELECT_0		Ram	00-1f	1f	TODO
TOP_PRE_INPUT_SELECT_1		Ram	00-1f	1f	TODO
TOP_PRE_INPUT_SELECT_2		Ram	00-1f	1f	TODO
TOP_PRE_INPUT_SELECT_3		Ram	00-1f	1f	TODO

2.3. Peripheral logic blocks

2.3.9 CMUXVR

The two Global Vertical CMUX drive 20 horizontal RCLK grids each half on each side. Ten are lost when touching the HPS.

Name	Instance	Type	Values	Default	Documentation
ENABLE_REGISTER_MODE	0-19	Mux	<ul style="list-style-type: none"> enout reg1_enout reg2_enout vcc 	vcc	TODO
ENABLE_REGISTER_POWER_UP	0-1	Num	<ul style="list-style-type: none"> 0-1 	1	TODO
INPUT_SELECT	0-19	Ram	0-f	b	TODO
PLL_FEEDBACK_ENABLE_0		Mux	<ul style="list-style-type: none"> vcc pll_mcnt0 	vcc	TODO

2.3.10 CTRL

The Control block gives access to a number of anciliary functions of the FPGA.

TODO: everything, GOUT/GIN/DCMUX mapping is done

2.3.11 HSSI

The High speed serial interface blocks control the serializing/deserializing capabilities of the FPGA.

TODO: everything

Name	Instance	Type	Values	Default	Documentation
PCS8G_AGGREGATE_DSKW_CONTROL		CTRL	<ul style="list-style-type: none"> write read 	write	TODO
PCS8G_AGGREGATE_DSKW_SM_OPERATION		OPERATION	<ul style="list-style-type: none"> xaui_sm srio_sm 	xaui_sm	TODO
PCS8G_AGGREGATE_PCS_DW_BONDING		BONDING	<ul style="list-style-type: none"> disable 	disable	TODO
PCS8G_AGGREGATE_POWERDOWN		DOWN	t/f	f	TODO
PCS8G_AGGREGATE_REFCLK_DISABLE		DISABLE	t/f	f	TODO

continues on next page

Table 8 – continued from previous page

Name	Instance	Type	Values	Default	Documentation
PCS8G_AGGREGATE_XAUI_SM		Mux	<ul style="list-style-type: none"> • xau_legacy_sm • xau_sm • disable 	xau_legacy_sm	TODO
COM_PCS_PLD_I0_2_HIP_EN		Bool	t/f	f	TODO
COM_PCS_PLD_I0_2_HRDRSTCTRL_B0_0_USR_EN		Bool	t/f	f	TODO
COM_PCS_PLD_I0_2_HRDRSTCTRL_B0_0_EN		Bool	t/f	f	TODO
COM_PCS_PLD_I0_2_TESTBUF_SEL		Mux	<ul style="list-style-type: none"> • pcs8g • pma_if 	pcs8g	TODO
COM_PCS_PLD_I0_2_USRMODE_SELMRST		Mux	<ul style="list-style-type: none"> • usermode • last_frz 	usermode	TODO
COM_PCS_PLD_I0_2_SIDE_RES_SMC0		Mux	<ul style="list-style-type: none"> • pld • b_hip 	pld	TODO
COM_PCS_PLD_I0_2_SIDE_RES_SMC1		Mux	<ul style="list-style-type: none"> • pld • b_hip 	pld	TODO
COM_PCS_PLD_I0_2_SIDE_RES_SMC10		Mux	<ul style="list-style-type: none"> • pld • b_hip 	pld	TODO
COM_PCS_PLD_I0_2_SIDE_RES_SMC11		Mux	<ul style="list-style-type: none"> • pld • b_hip 	pld	TODO
COM_PCS_PLD_I0_2_SIDE_RES_SMC12		Mux	<ul style="list-style-type: none"> • pld • b_hip 	pld	TODO
COM_PCS_PLD_I0_2_SIDE_RES_SMC13		Mux	<ul style="list-style-type: none"> • pld • b_hip 	pld	TODO
COM_PCS_PLD_I0_2_SIDE_RES_SMC14		Mux	<ul style="list-style-type: none"> • pld • b_hip 	pld	TODO
COM_PCS_PLD_I0_2_SIDE_RES_SMC15		Mux	<ul style="list-style-type: none"> • pld • b_hip 	pld	TODO

continues on next page

Table 8 – continued from previous page

Name	Instance	Type	Values	Default	Documentation
COM_PCS_PLD	PLD_SIDE_RES_SMC6	SMC6	<ul style="list-style-type: none"> • pld • b_hip 	pld	TODO
COM_PCS_PLD	PLD_SIDE_RES_SMC7	SMC7	<ul style="list-style-type: none"> • pld • b_hip 	pld	TODO
COM_PCS_PLD	PLD_SIDE_RES_SMC8	SMC8	<ul style="list-style-type: none"> • pld • b_hip 	pld	TODO
COM_PCS_PLD	PLD_SIDE_RES_SMC9	SMC9	<ul style="list-style-type: none"> • pld • b_hip 	pld	TODO
COM_PCS_PLD	SIDE_DATA_SRC	Mux	<ul style="list-style-type: none"> • pld • b_hip 	pld	TODO
COM_PCS_PMA	IF2AUTO_SPEED_EN	Bool	t/f	f	TODO
COM_PCS_PMA	IF2BLOCK_SEL	Bool	t/f	f	TODO
COM_PCS_PMA	IF2FORCE_FREQDET	Bool	<ul style="list-style-type: none"> • off • force0 • force1 	off	TODO
COM_PCS_PMA	IF2G3PCS	Bool	t/f	f	TODO
COM_PCS_PMA	IF2PMA_IF_DFT_BNol	Bool	t/f	f	TODO
COM_PCS_PMA	IF2PMA_IF_DFT_RAln	Bool	0-1	0	TODO
COM_PCS_PMA	IF2PM_GEN1_2_CNTx	CNTx	<ul style="list-style-type: none"> • cnt_32k • cnt_64k 	cnt_32k	TODO
COM_PCS_PMA	IF2PPMSEL	Mux	<ul style="list-style-type: none"> • default • ppm_100 • ppm_125 • ppm_62_5 • ppm_200 • ppm_300 • ppm_250 • ppm_500 • ppm_1000 • ppm_other 	default	TODO
COM_PCS_PMA	IF2PPM_CNT_RST	Bool	t/f	f	TODO

continues on next page

Table 8 – continued from previous page

Name	Instance	Type	Values	Default	Documentation
COM_PCS_PMA	IF2PPM_EARLY	DEASSERT	t/f	f	TODO
COM_PCS_PMA	IF2PPM_POST_E	INLH_DLY	<ul style="list-style-type: none"> • 200 • 400 	200	TODO
PCS8G_BASE_ADDR		Ram	000-7ff		TODO
PCS8G_DEFAULT_BROADCAST_EN		Bool	t/f	f	TODO
PCS8G_DIGI_RX_0-22_SYMBOL_BCR		Ram	000-fff	0	TODO
PCS8G_DIGI_RX_0-3B10B_DECODER		Mux	<ul style="list-style-type: none"> • off • sgx • ibm 	off	TODO
PCS8G_DIGI_RX_0-3B10B_DECODER		OUTPUT_SEL	<ul style="list-style-type: none"> • data_8b10b • data_xaui_sm 	data_8b10b	TODO
PCS8G_DIGI_RX_0-3AEC_BLOCK_SEL		Mux	<ul style="list-style-type: none"> • same • other 	same	TODO
PCS8G_DIGI_RX_0-3A2TO_ERROR_REPLACE_EN		Bool	t/f	f	TODO
PCS8G_DIGI_RX_0-3A2TO_SPEED_NEG		Bool	40 bits	0	TODO
PCS8G_DIGI_RX_0-3BDS_DEC_CLOCK_GATING_EN		Bool	t/f	f	TODO
PCS8G_DIGI_RX_0-3BST_CLOCK_GATING_EN		Bool	t/f	f	TODO
PCS8G_DIGI_RX_0-3BST_CLR_FLAG_EN		Bool	t/f	f	TODO
PCS8G_DIGI_RX_0-3BST_VER		Mux	<ul style="list-style-type: none"> • disable • incremental • cjpat • crpat 	disable	TODO
PCS8G_DIGI_RX_0-3BT_REVERSAL_EN		Bool	t/f	f	TODO
PCS8G_DIGI_RX_0-3BYTEORDER_CLOCK_GATING_EN		Bool	t/f	f	TODO
PCS8G_DIGI_RX_0-3BYTE_DESERIALIZER		Mux	<ul style="list-style-type: none"> • disable • bds_by_2 • bds_by_2_det 	disable	TODO
PCS8G_DIGI_RX_0-3BYTE_ORDER		Ram	23 bits	0	TODO
PCS8G_DIGI_RX_0-3CDR_CTRL		Ram	30 bits	0	TODO
PCS8G_DIGI_RX_0-3CEIFO_RST_PLD_CTRL_EN		Bool	t/f	f	TODO
PCS8G_DIGI_RX_0-3CD_PATTERN		Ram	00-ff	0	TODO

continues on next page

Table 8 – continued from previous page

Name	Instance	Type	Values	Default	Documentation
PCS8G_DIGI_RX_CLK1		Mux	<ul style="list-style-type: none"> clk1 tx_pma agg agg_top_or_bottom 	clk1	TODO
PCS8G_DIGI_RX_CLK2		Mux	<ul style="list-style-type: none"> rcvd_clk tx_pma refclk_dig2 	rcvd_clk	TODO
PCS8G_DIGI_RX_CLK_FREE_RUNNING_EN		Bool	t/f	f	TODO
PCS8G_DIGI_RX_DESKEW		Mux	<ul style="list-style-type: none"> disable xau1 srio_v2p1 	disable	TODO
PCS8G_DIGI_RX_DESKEW_PROG_BA1_ONLY_EN		Bool	t/f	f	TODO
PCS8G_DIGI_RX_DESKEW_RDCLK_GATEING_EN		Bool	t/f	f	TODO
PCS8G_DIGI_RX_DW_DESKEW_WRCLK_GATEING_EN		Bool	t/f	f	TODO
PCS8G_DIGI_RX_DW_PC_WRCLK_GATEING_EN		Bool	t/f	f	TODO
PCS8G_DIGI_RX_DW_RM_RDCLK_GATEING_EN		Bool	t/f	f	TODO
PCS8G_DIGI_RX_DW_RM_WRCLK_GATEING_EN		Bool	t/f	f	TODO
PCS8G_DIGI_RX_DW_WA_CLOCK_GATEING_EN		Bool	t/f	f	TODO
PCS8G_DIGI_RX_IDLE_CLOCK_GATEING_EN		Bool	t/f	f	TODO
PCS8G_DIGI_RX_IDLE_EIOS_EN		Bool	t/f	f	TODO
PCS8G_DIGI_RX_IDLE_ENTRY_IDLE_EN		Bool	t/f	f	TODO
PCS8G_DIGI_RX_IDLE_ENTRY_SLEEP_EN		Bool	t/f	f	TODO
PCS8G_DIGI_RX_ERR_FLAGS_SEL		Mux	<ul style="list-style-type: none"> flags_8b10b flags_wa 	flags_8b10b	TODO
PCS8G_DIGI_RX_INVALID_CODE_FLAG_ONLY_EN		Bool	t/f	f	TODO
PCS8G_DIGI_RX_PAD_EDB_ERROR_REPLACE		Bool	<ul style="list-style-type: none"> edb pad edb_dynamic 	edb	TODO
PCS8G_DIGI_RX_PARALLEL_LOOPBACK_EN		Bool	t/f	f	TODO
PCS8G_DIGI_RX_PCIE_FIFO_RST_PLD_CTRL_EN		Bool	t/f	f	TODO
PCS8G_DIGI_RX_PCS_BYPASS_EN		Bool	t/f	f	TODO
PCS8G_DIGI_RX_PCS_URST_EN		Bool	t/f	f	TODO
PCS8G_DIGI_RX_PCIE_RDCLK_GATEING_EN		Bool	t/f	f	TODO

continues on next page

Table 8 – continued from previous page

Name	Instance	Type	Values	Default	Documentation
PCS8G_DIGI_RX_PIPELINE_COMPENSATION_FIFO	PIPELINE_COMPENSATION_FIFO	Stat	<ul style="list-style-type: none"> normal_latency pid_ctrl_normal_latency low_latency pid_ctrl_low_latency register_fifo 	normal_latency	TODO
PCS8G_DIGI_RX_PIPE_IF_EN	PIPE_IF_EN	Bool	t/f	f	TODO
PCS8G_DIGI_RX_PLANE_BONDING_COMP_EN	PLANE_BONDING_COMP_EN	Bool	t/f	f	TODO
PCS8G_DIGI_RX_PLANE_BONDING_MASTER	PLANE_BONDING_MASTER	Bool	t/f	f	TODO
PCS8G_DIGI_RX_PMA_DW	PMA_DW	Num	<ul style="list-style-type: none"> 8 10 16 20 	8	TODO
PCS8G_DIGI_RX_POLARITY_INVERSION_EN	POLARITY_INVERSION_EN	Bool	t/f	f	TODO
PCS8G_DIGI_RX_POLINV_8B10B_DEC_EN	POLINV_8B10B_DEC_EN	Bool	t/f	f	TODO
PCS8G_DIGI_RX_PRBS_CLOCK_GATEING_EN	PRBS_CLOCK_GATEING_EN	Bool	t/f	f	TODO
PCS8G_DIGI_RX_PRBS_CLR_FLACEN	PRBS_CLR_FLACEN	Bool	t/f	f	TODO
PCS8G_DIGI_RX_PRBS_VER	PRBS_VER	Mux	<ul style="list-style-type: none"> disable prbs_7_dw_8_10 prbs_23_dw_hf_sw prbs_7_sw_hf_dw_lf_sw prbs_lf_dw_mf_sw prbs_23_sw_mf_dw prbs_15 prbs_31 	disable	TODO
PCS8G_DIGI_RX_RATHER_MATCH	RATHER_MATCH	Ram	68 bits	0	TODO
PCS8G_DIGI_RX_RECV_CLK	REVD_CLK	Mux	<ul style="list-style-type: none"> rcvd_clk tx_pma 	rcvd_clk	TODO
PCS8G_DIGI_RX_RD_CLK	RD_CLK	Mux	<ul style="list-style-type: none"> rx_clk pld 	rx_clk	TODO
PCS8G_DIGI_RX_REFCLK_SEL_EN	REFCLK_SEL_EN	Bool	t/f	f	TODO

continues on next page

Table 8 – continued from previous page

Name	Instance	Type	Values	Default	Documentation
PCS8G_DIGI_RX_02A_BO_ON_WA_BN01	02A_BO_ON_WA_BN01	Bool	t/f	f	TODO
PCS8G_DIGI_RX_02A_LENGTH_CHECK	02A_LENGTH_CHECK	Bool	00-7f	0	TODO
PCS8G_DIGI_RX_02A_DESKEW_WRCLK_GATE_EN	02A_DESKEW_WRCLK_GATE_EN	Bool	t/f	f	TODO
PCS8G_DIGI_RX_02A_PC_WRCLK_GATE_EN	02A_PC_WRCLK_GATE_EN	Bool	t/f	f	TODO
PCS8G_DIGI_RX_02A_RM_RDCLK_GATE_EN	02A_RM_RDCLK_GATE_EN	Bool	t/f	f	TODO
PCS8G_DIGI_RX_02A_RM_WRCLK_GATE_EN	02A_RM_WRCLK_GATE_EN	Bool	t/f	f	TODO
PCS8G_DIGI_RX_02A_SYMBOL_SWAP_EN01	02A_SYMBOL_SWAP_EN01	Bool	t/f	f	TODO
PCS8G_DIGI_RX_02A_TEST_BUS_SEL	02A_TEST_BUS_SEL	Mux	<ul style="list-style-type: none"> prbs_bist tx tx_ctrl_plane wa deskew rm rx_ctrl pcie_ctrl rx_ctrl_plane agg 	prbs_bist	TODO
PCS8G_DIGI_RX_02A_VALID_MASK_EN01	02A_VALID_MASK_EN01	Bool	t/f	f	TODO
PCS8G_DIGI_RX_02A_BOUNDARY_LOCK	02A_BOUNDARY_LOCK	Bool	<ul style="list-style-type: none"> auto_align_pld_ctrl sync_sm deterministic_latency bit_slip 	auto_align_pld_ctrl	TODO
PCS8G_DIGI_RX_02A_CLK_SLIP_SRAGING	02A_CLK_SLIP_SRAGING	Bool	000-3ff	0	TODO
PCS8G_DIGI_RX_02A_CLOCK_GATE_EN	02A_CLOCK_GATE_EN	Bool	t/f	f	TODO
PCS8G_DIGI_RX_02A_DET_LATENCY_SYNC_STATUS	02A_DET_LATENCY_SYNC_STATUS	Bool	<ul style="list-style-type: none"> delayed immediate 	delayed	TODO
PCS8G_DIGI_RX_02A_DISP_ERR_FLAG_EN	02A_DISP_ERR_FLAG_EN	Bool	t/f	f	TODO
PCS8G_DIGI_RX_02A_KCHAR_EN	02A_KCHAR_EN	Bool	t/f	f	TODO
PCS8G_DIGI_RX_02A_PD	02A_PD	Ram	43 bits	0	TODO
PCS8G_DIGI_RX_02A_PLD_CONTROLLED	02A_PLD_CONTROLLED	Bool	<ul style="list-style-type: none"> level_sensitive pid_ctrl_sw rising_edge_sensitive 	level_sensitive	TODO
PCS8G_DIGI_RX_02A_SYNC_SM_CTRL	02A_SYNC_SM_CTRL	Ctrl	38 bits	0	TODO

continues on next page

Table 8 – continued from previous page

Name	Instance	Type	Values	Default	Documentation
PCS8G_DIGI_RX_OVR_CLK		Mux	<ul style="list-style-type: none"> rx_clk2 txfifo_rd_clk 	rx_clk2	TODO
PCS8G_DIGI_TX_GB10B_DISP_CTRL		Mux	<ul style="list-style-type: none"> off on_ib on 	off	TODO
PCS8G_DIGI_TX_GB10B_ENCODER		Mux	<ul style="list-style-type: none"> off ibm sgx 	off	TODO
PCS8G_DIGI_TX_GB10B_ENCODER_INPUT			<ul style="list-style-type: none"> xau_i_sm normal_data_path gige_idle_conversion 	xau_i_sm	TODO
PCS8G_DIGI_TX_AGC_BLOCK_SEL		Mux	<ul style="list-style-type: none"> same other 	same	TODO
PCS8G_DIGI_TX_BEST_CLOCK_GATE_EN		Bool	t/f	f	TODO
PCS8G_DIGI_TX_BEST_GEN		Mux	<ul style="list-style-type: none"> disable incremental cjpat crpat 	disable	TODO
PCS8G_DIGI_TX_BETSLIP_EN		Bool	t/f	f	TODO
PCS8G_DIGI_TX_BIT_REVERSAL_EN		Bool	t/f	f	TODO
PCS8G_DIGI_TX_B3_CLOCK_GATE_EN		Bool	t/f	f	TODO
PCS8G_DIGI_TX_BYPASS_PIPELINE_REG_EN		Bool	t/f	f	TODO
PCS8G_DIGI_TX_BYTE_SERIALIZER_EN		Bool	t/f	f	TODO
PCS8G_DIGI_TX_CC_DISPARITY_EN		Bool	t/f	f	TODO
PCS8G_DIGI_TX_CD_PATTERN		Ram	000-1ff	0	TODO
PCS8G_DIGI_TX_DYNAMIC_CLOCK_SWITCH_EN		Bool	t/f	f	TODO
PCS8G_DIGI_TX_FIFORD_CLOCK_GATE_EN		Bool	t/f	f	TODO
PCS8G_DIGI_TX_FIFOWR_CLOCK_GATE_EN		Bool	t/f	f	TODO
PCS8G_DIGI_TX_FORCE_ECHAR_EN		Bool	t/f	f	TODO
PCS8G_DIGI_TX_FORCE_KCHAR_EN		Bool	t/f	f	TODO
PCS8G_DIGI_TX_G2_FREQUENCY_SCALING			<ul style="list-style-type: none"> off on 	off	TODO

continues on next page

Table 8 – continued from previous page

Name	Instance	Type	Values	Default	Documentation
PCS8G_DIGI_TX_OOOPBACK		Bool	t/f	f	TODO
PCS8G_DIGI_TX_PCFIFO_URST_EN		Bool	t/f	f	TODO
PCS8G_DIGI_TX_PCS_BYPASS_EN		Bool	t/f	f	TODO
PCS8G_DIGI_TX_PHASE_COMPENSATION_FIFO			<ul style="list-style-type: none"> normal_latency pid_ctrl_normal_latency low_latency pid_ctrl_low_latency register_fifo 	normal_latency	TODO
PCS8G_DIGI_TX_PCFIFO_REFCLK_MUX_SEL			<ul style="list-style-type: none"> refclk tx_pma 	refclk	TODO
PCS8G_DIGI_TX_PCFIFO_WRITE_CLK_SEL			<ul style="list-style-type: none"> pld tx_clk 	pld	TODO
PCS8G_DIGI_TX_PLANE_BONDING_COMP_EN		Bool	t/f	f	TODO
PCS8G_DIGI_TX_PLANE_BONDING_CONSUMPTION			<ul style="list-style-type: none"> individual bundled_master slave_above slave_below 	individual	TODO
PCS8G_DIGI_TX_PLANE_BONDING_CONSUMPTION			<ul style="list-style-type: none"> individual bundled_master slave_above slave_below 	individual	TODO
PCS8G_DIGI_TX_PLANE_BONDING_MASTER		Bool	t/f	f	TODO
PCS8G_DIGI_TX_PMA_DW		Num	<ul style="list-style-type: none"> 8 10 16 20 	8	TODO
PCS8G_DIGI_TX_POLARITY_INVERSION_EN		Bool	t/f	f	TODO

continues on next page

Table 8 – continued from previous page

Name	Instance	Type	Values	Default	Documentation
PCS8G_DIGI_TX_0PRBS_CLOCK_GATE_EN		Bool	t/f	f	TODO
PCS8G_DIGI_TX_0PRBS_GEN		Mux	<ul style="list-style-type: none"> • disable • prbs_7_dw_8_10 • prbs_23_dw_hf_sw • prbs_7_sw_hf_dw_lf_sw • prbs_lf_dw_mf_sw • prbs_23_sw_mf_dw • prbs_15 • prbs_31 	disable	TODO
PCS8G_DIGI_TX_0SYMBOL_SWAP_EN		Bool	t/f	f	TODO
PCS8G_DIGI_TX_0TXCLK_FREERUN_EN		Bool	t/f	f	TODO
PCS8G_DIGI_TX_0TXPCS_URST_EN		Bool	t/f	f	TODO
PCS8G_MDIO_0DIS_CVP_EN		Bool	t/f	f	TODO
PCS8G_MDIO_0DIS_FORCE_EN		Bool	t/f	f	TODO
PCS8G_PIPE_INTB_0TOP_DESERIALIZE_EN		Bool	t/f	f	TODO
PCS8G_PIPE_INTB_0TOP_ERROR_REPLACE_PAD		Mux	<ul style="list-style-type: none"> • edb • pad 	edb	TODO
PCS8G_PIPE_INTB_0TOP_IND_ERROR_REPORTING		Bool	t/f	f	TODO
PCS8G_PIPE_INTB_0TOP_PHYSTATUS_RST_TOGGLE		Bool	t/f	f	TODO
PCS8G_PIPE_INTB_0TOP_RPRE_EMULATION_SETTINGS		30 bits	0	0	TODO
PCS8G_PIPE_INTB_0TOP_RVOD_SERIALIZATION_SETTINGS		30 bits	0	0	TODO
PCS8G_PIPE_INTB_0TOP_RXDETECT_BYPASS_EN		Bool	t/f	f	TODO
PCS8G_PIPE_INTB_0TOP_RX_PIPE_ENABLE		Bool	t/f	f	TODO
PCS8G_PIPE_INTB_0TOP_TXSWING_ENABLE		Bool	t/f	f	TODO
PCS8G_PIPE_INTB_0TOP_TX_PIPE_ENABLE		Bool	t/f	f	TODO
PCS8G_POWER_0ISOLATION_EN		Bool	t/f	f	TODO
PCS9G_PIPE_INTB_0TOP_ELECIDLER_DELAY		0-7	0	0	TODO
PCS9G_PIPE_INTB_0TOP_PHY_STATUS_DELAY		0-7	0	0	TODO
PLD_PCS_DEFAULT_02_BROADCAST_ENABLE		Bool	t/f	f	TODO
PLD_PCS_IF_BASE_02_ADDR		Ram	000-7ff		TODO
PLD_PCS_MDIO_0DIS_CVP_EN		Bool	t/f	f	TODO
PLD_PCS_MDIO_0DIS_FORCE_EN		Bool	t/f	f	TODO
PLD_PCS_POWER_0ISOLATION_EN		Bool	t/f	f	TODO
PMA_PCS_DEFAULT_01_BROADCAST_ENABLE		Bool	t/f	f	TODO
PMA_PCS_IF_BASE_02_ADDR		Ram	000-7ff		TODO
PMA_PCS_MDIO_0DIS_CVP_EN		Bool	t/f	f	TODO
PMA_PCS_MDIO_0DIS_FORCE_EN		Bool	t/f	f	TODO
PMA_PCS_POWER_0ISOLATION_EN		Bool	t/f	f	TODO

continues on next page

Table 8 – continued from previous page

Name	Instance	Type	Values	Default	Documentation
RX_PCS_PLD_IF0_PCS_SIDE_BLOCK_SEL		Mux	<ul style="list-style-type: none"> • default • pcs8g 	default	TODO
RX_PCS_PLD_SIDE2_DATA_SRC		Mux	<ul style="list-style-type: none"> • pld • b_hip 	pld	TODO
RX_PCS_PMA_IF0-2		Mux	<ul style="list-style-type: none"> • default • pcs8g 	default	TODO
RX_PCS_PMA_IF0_CLKSLIP_SEL		Mux	<ul style="list-style-type: none"> • pld • slip_pcs8g 	pld	TODO
TX_PCS_PLD_SIDE2_DATA_SRC		Mux	<ul style="list-style-type: none"> • pld • b_hip 	pld	TODO
TX_PCS_PMA_IF0_BLOCK_SEL		Mux	<ul style="list-style-type: none"> • default • pcs8g 	default	TODO

2.3.12 HIP

The PCIe Hard-IP blocks control the PCIe interfaces of the FPGA.

TODO: everything

Name	Instance	Type	Values	Default	Documentation
BIST_MEMORY_SETTINGS_DATA		Ram	75 bits	0	TODO
BRIDGE_66MHZCAP		Bool	t/f	f	TODO
BR_RCB		Mux	<ul style="list-style-type: none"> • ro • rw 	ro	TODO
BYPASS_CDC		Bool	t/f	f	TODO
BYPASS_CLK_SWITCH		Bool	t/f	f	TODO
BYPASS_TL		Bool	t/f	f	TODO
CDC_CLK_RELATION		Mux	<ul style="list-style-type: none"> • plesiochronous • mesochronous 	plesiochronous	TODO
CDC_DUMMY_INSERT_LIMIT_DATA		Ram	0-f	0	TODO

continues on next page

Table 9 – continued from previous page

Name	Instance	Type	Values	Default	Documentation
CORE_CLK_DISABLE_CLK_SWITCH		Mux	<ul style="list-style-type: none"> core_clk_out pld_clk 	core_clk_out	TODO
CORE_CLK_DIVIDER		Num	<ul style="list-style-type: none"> 1-2 4 8 16 	4	TODO
CORE_CLK_OUT_SEL		Mux	<ul style="list-style-type: none"> div_1 div_2 	div_1	TODO
CORE_CLK_SEL		Mux	<ul style="list-style-type: none"> core_clk_out pld_clk 	core_clk_out	TODO
CORE_CLK_SOURCE		Mux	<ul style="list-style-type: none"> pll_fixed_clk core_clk_in pclk_in 	pll_fixed_clk	TODO
CVP_CLK_RESET		Bool	t/f	f	TODO
CVP_DATA_COMPRESSED		Bool	t/f	f	TODO
CVP_DATA_ENCRYPTED		Bool	t/f	f	TODO
CVP_ISOLATION		Bool	t/f	f	TODO
CVP_MODE_RESET		Bool	t/f	f	TODO
CVP_RATE_SEL		Mux	<ul style="list-style-type: none"> full_rate half_rate 	full_rate	TODO
DEVICE_NUMBER_DATA		Ram	00-1f	0	TODO
DEVSELTIM		Mux	<ul style="list-style-type: none"> fast_devsel_decoding medium_devsel_decoding slow_devsel_decoding 	fast_devsel_decoding	TODO
DISABLE_AUTO_CRCS		Bool	t/f	f	TODO
DISABLE_CLK_SWITCH		Bool	t/f	f	TODO
DISABLE_LINK_X2_SUPPORT		Bool	t/f	f	TODO
DISABLE_TAG_CHECK		Bool	t/f	f	TODO
EI_DELAY_POWERDOWN_COUNTER		TRDATA	00-ff	0	TODO

continues on next page

Table 9 – continued from previous page

Name	Instance	Type	Values	Default	Documentation
ENABLE_ADAPTER_HALF_RATE_MODE		Bool	t/f	f	TODO
ENABLE_CH01_PCLK_OUT		Mux	<ul style="list-style-type: none"> • pclk_ch0 • pclk_ch1 	pclk_ch0	TODO
ENABLE_CH0_PCLK_OUT		Mux	<ul style="list-style-type: none"> • pclk_central • pclk_ch01 	pclk_central	TODO
ENABLE_RX_BUFFER_CHECKING		Bool	t/f	f	TODO
ENABLE_RX_REORDERING		Bool	t/f	f	TODO
FASTB2BCAP		Bool	t/f	f	TODO
FC_INIT_TIMER_DATA		Ram	000-7ff	0	TODO
FLOW_CONTROL_TIMEOUT_COUNTER_DATA		Ram	00-ff	0	TODO
FLOW_CONTROL_UPDATE_COUNTER_DATA		Ram	00-1f	0	TODO
GEN12_LANE_RATE_MODE		Mux	<ul style="list-style-type: none"> • gen1 • gen1_gen2 	gen1	TODO
HARD_RESET_BYPASS		Bool	t/f	f	TODO
IEI_ENABLE_SETTINGS		Mux	<ul style="list-style-type: none"> • disabled • disable_iei_logic • gen2_infei_gen1_infei • gen2_infei_gen1_infei_sd • gen2_infei_infsd_gen1_infei_sd • gen2_infei_infsd_gen1_infei_infsd 	disabled	TODO
JTAG_ID_DATA		Ram	128 bits	0	TODO
L01_ENTRY_LATENCY_DATA		Ram	00-1f	0	TODO
LANE_MASK		Mux	<ul style="list-style-type: none"> • x8 • x1 • x2 • x4 	x8	TODO
LATTIM_RO_DATA		Ram	00-7f	0	TODO
MDIO_CB_OPBIT_ENABLE		Bool	t/f	f	TODO
MEMWRINV		Mux	<ul style="list-style-type: none"> • ro • rw 	ro	TODO

continues on next page

Table 9 – continued from previous page

Name	Instance	Type	Values	Default	Documentation
MILLISECOND_CYCLE_COUNT_DATA		Num	20 bits	0	TODO
MULTI_FUNCTION		Num	<ul style="list-style-type: none"> • 1-8 	1	TODO
NATIONAL_INST_THRU_ENHANCE		Bool	t/f	f	TODO
PCIE_MODE		Mux	<ul style="list-style-type: none"> • ep_native • ep_legacy • rp • sw_up • sw_dn • bridge • switch_mode • shared_mode 	ep_native	TODO
PCIE_SPEC_1P0_COMPLIANCE		Mux	<ul style="list-style-type: none"> • spec_1p0a • spec_1p1 	spec_1p0a	TODO
PCLK_OUT_SEL		Mux	<ul style="list-style-type: none"> • core_clk_en • pclk_out 	core_clk_en	TODO
PIPEX1_DEBUG_SEL		Bool	t/f	f	TODO
PLNIOTRI_GATE		Bool	t/f	f	TODO
PORT_LINK_NUMBER_DATA		Ram	00-ff	0	TODO
REGISTER_PIPE_SIGNALS		Bool	t/f	f	TODO
RETRY_BUFFER_LAST_ACTIVE_ADDRESS_DATA		Num	00-ff	0	TODO
RETRY_BUFFER_MEMORY_SETTINGS_DATA		Num	0000-ffff	0	TODO
RSTCTRL_1MS_COUNT_FREF_CLK_VALUE		Ram	20 bits	0	TODO
RSTCTRL_1US_COUNT_FREF_CLK_VALUE		Ram	20 bits	0	TODO
RSTCTRL_ALTP2_CRST_N_INV		Bool	t/f	f	TODO
RSTCTRL_ALTP2_RST_N_INV		Bool	t/f	f	TODO
RSTCTRL_ALTP2_SRST_N_INV		Bool	t/f	f	TODO
RSTCTRL_DEBUG_EN		Bool	t/f	f	TODO
RSTCTRL_FORCE_INACTIVE_RST		Bool	t/f	f	TODO

continues on next page

Table 9 – continued from previous page

Name	Instance	Type	Values	Default	Documentation
RSTCTRL_FREF_CLK_SELECT		Mux	<ul style="list-style-type: none"> disabled ch0_sel ch1_sel ch2_sel ch3_sel ch4_sel ch5_sel ch6_sel ch7_sel ch8_sel ch9_sel ch10_sel ch11_sel 	disabled	TODO
RSTCTRL_HARD_BLOCK_ENABLE		Mux	<ul style="list-style-type: none"> hard_rst_ctl pld_rst_ctl 	hard_rst_ctl	TODO
RSTCTRL_HIP_EP		Mux	<ul style="list-style-type: none"> hip_not_ep hip_ep 	hip_not_ep	TODO
RSTCTRL_LTSSM_DISABLE		Bool	t/f	f	TODO
RSTCTRL_MASK_TX_PLL_LOCK_SELECT		Mux	<ul style="list-style-type: none"> disabled ch1_sel ch4_sel ch4_10_sel 	disabled	TODO
RSTCTRL_OFF_CAL_DONE_SELECT		Mux	<ul style="list-style-type: none"> disabled ch0_out ch01_out ch0123_out ch0123_5678_out 	disabled	TODO

continues on next page

Table 9 – continued from previous page

Name	Instance	Type	Values	Default	Documentation
RSTCTRL_OFF_CAL_EN_SELECT		Mux	<ul style="list-style-type: none"> disabled ch0_out ch01_out ch0123_out ch0123_5678_out 	disabled	TODO
RSTCTRL_PERSTN_SELECT		Mux	<ul style="list-style-type: none"> perstn_pin perstn_pld 	perstn_pin	TODO
RSTCTRL_PERST_ENABLE		Mux	<ul style="list-style-type: none"> level neg_edge 	level	TODO
RSTCTRL_PLD_CLR		Bool	t/f	f	TODO
RSTCTRL_RX_PCS_RST_N_INV		Bool	t/f	f	TODO
RSTCTRL_RX_PCS_RST_N_SELECT		Mux	<ul style="list-style-type: none"> disabled ch0_out ch01_out ch0123_out ch012345678_out ch012345678_10_out 	disabled	TODO
RSTCTRL_RX_PLL_FREQ_LOCK_SELECT		Mux	<ul style="list-style-type: none"> disabled ch0_sel ch01_sel ch0123_sel ch0123_5678_sel ch0123_5678_phs_sel ch0123_phs_sel ch01_phs_sel ch0_phs_sel 	disabled	TODO

continues on next page

Table 9 – continued from previous page

Name	Instance	Type	Values	Default	Documentation
RSTCTRL_RX_PLL_LOCK_SELECT		Mux	<ul style="list-style-type: none"> disabled ch0_sel ch01_sel ch0123_sel ch0123_5678_sel 	disabled	TODO
RSTCTRL_RX_PMA_RSTB_CMU_SELECT		Mux	<ul style="list-style-type: none"> disabled ch1cmu_sel ch4cmu_sel ch4_10cmu_sel 	disabled	TODO
RSTCTRL_RX_PMA_RSTB_INV		Bool	t/f	f	TODO
RSTCTRL_RX_PMA_RSTB_SELECT		Mux	<ul style="list-style-type: none"> disabled ch0_out ch01_out ch0123_out ch012345678_out ch012345678_10_out 	disabled	TODO
RSTCTRL_TIMER_A_TYPE		Mux	<ul style="list-style-type: none"> disabled milli_secs micro_secs fref_cycles 	disabled	TODO
RSTCTRL_TIMER_A_VALUE		Ram	00-ff	0	TODO
RSTCTRL_TIMER_B_TYPE		Mux	<ul style="list-style-type: none"> disabled milli_secs micro_secs fref_cycles 	disabled	TODO
RSTCTRL_TIMER_B_VALUE		Ram	00-ff	0	TODO

continues on next page

Table 9 – continued from previous page

Name	Instance	Type	Values	Default	Documentation
RSTCTRL_TIMER_C_TYPE		Mux	<ul style="list-style-type: none"> • disabled • milli_secs • micro_secs • fref_cycles 	disabled	TODO
RSTCTRL_TIMER_C_VALUE		Ram	00-ff	0	TODO
RSTCTRL_TIMER_D_TYPE		Mux	<ul style="list-style-type: none"> • disabled • milli_secs • micro_secs • fref_cycles 	disabled	TODO
RSTCTRL_TIMER_D_VALUE		Ram	00-ff	0	TODO
RSTCTRL_TIMER_E_TYPE		Mux	<ul style="list-style-type: none"> • disabled • milli_secs • micro_secs • fref_cycles 	disabled	TODO
RSTCTRL_TIMER_E_VALUE		Ram	00-ff	0	TODO
RSTCTRL_TIMER_F_TYPE		Mux	<ul style="list-style-type: none"> • disabled • milli_secs • micro_secs • fref_cycles 	disabled	TODO
RSTCTRL_TIMER_F_VALUE		Ram	00-ff	0	TODO
RSTCTRL_TIMER_G_TYPE		Mux	<ul style="list-style-type: none"> • disabled • milli_secs • micro_secs • fref_cycles 	disabled	TODO
RSTCTRL_TIMER_G_VALUE		Ram	00-ff	0	TODO

continues on next page

Table 9 – continued from previous page

Name	Instance	Type	Values	Default	Documentation
RSTCTRL_TIMER_H_TYPE		Mux	<ul style="list-style-type: none"> disabled milli_secs micro_secs fref_cycles 	disabled	TODO
RSTCTRL_TIMER_H_VALUE		Ram	00-ff	0	TODO
RSTCTRL_TIMER_I_TYPE		Mux	<ul style="list-style-type: none"> disabled milli_secs micro_secs fref_cycles 	disabled	TODO
RSTCTRL_TIMER_I_VALUE		Ram	00-ff	0	TODO
RSTCTRL_TIMER_J_TYPE		Mux	<ul style="list-style-type: none"> disabled milli_secs micro_secs fref_cycles 	disabled	TODO
RSTCTRL_TIMER_J_VALUE		Ram	00-ff	0	TODO
RSTCTRL_TX_CMU_PLL_LOCK_SELECT		Mux	<ul style="list-style-type: none"> disabled ch1_sel ch4_sel ch4_10_sel 	disabled	TODO
RSTCTRL_TX_LC_PLL_LOCK_SELECT		Mux	<ul style="list-style-type: none"> disabled ch1_sel ch7_sel 	disabled	TODO
RSTCTRL_TX_LC_PLL_RSTB_SELECT		Mux	<ul style="list-style-type: none"> disabled ch1_out ch7_out 	disabled	TODO
RSTCTRL_TX_PCS_RST_N_INV		Bool	t/f	f	TODO

continues on next page

Table 9 – continued from previous page

Name	Instance	Type	Values	Default	Documentation
RSTCTRL_TX_PCS_RST_N_SELECT		Mux	<ul style="list-style-type: none"> disabled ch0_out ch01_out ch0123_out ch012345678_out ch012345678_10_out 	disabled	TODO
RSTCTRL_TX_PMA_RSTB_INV		Bool	t/f	f	TODO
RSTCTRL_TX_PMA_SYNC_P_INV		Bool	t/f	f	TODO
RSTCTRL_TX_PMA_SYNC_P_SELECT		Mux	<ul style="list-style-type: none"> disabled ch1_out ch4_out ch4_10_out 	disabled	TODO
RXFREQCLK_CNT_DATA		Ram	20 bits	0	TODO
RXFREQCLK_CNT_EN		Bool	t/f	f	TODO
RX_CDC_ALMOST_FULL_DATA		Ram	0-f	0	TODO
RX_L0S_COUNT_IDL_DATA		Ram	00-ff	0	TODO
RX_PTR0_NONPOSTED_DPRAM_MAX_DATA		Ram	000-3ff	0	TODO
RX_PTR0_NONPOSTED_DPRAM_MIN_DATA		Ram	000-3ff	0	TODO
RX_PTR0_POSTED_DPRAM_MAX_DATA		Ram	000-3ff	0	TODO
RX_PTR0_POSTED_DPRAM_MIN_DATA		Ram	000-3ff	0	TODO
SINGLE_RX_DETECT_DATA		Ram	0-f	0	TODO
SKP_INSERTION_CONTROL		Bool	t/f	f	TODO
SKP_OS_SCHEDULE_COUNT_DATA		Ram	000-7ff	0	TODO
SLOTCLK_CFG		Mux	<ul style="list-style-type: none"> dynamic_slotclkcfg static_slotclkcfgoff static_slotclkcfgon 	dynamic_slotclkcfg	TODO
SLOT_REGISTER_EN		Bool	t/f	f	TODO
TESTMODE_CONTROL		Bool	t/f	f	TODO
TX_CDC_ALMOST_FULL_DATA		Ram	0-f	0	TODO
TX_L0S_ADJUST		Bool	t/f	f	TODO
TX_SWING_DATA		Ram	00-ff	0	TODO
USER_ID_DATA		Ram	0000-ffff	0	TODO
USE_CRC_FORWARDING		Bool	t/f	f	TODO
VC0_CLK_ENABLE		Bool	t/f	f	TODO
VC0_RX_BUFFER_MEMORY_SETTINGS_DATA		Ram	0000-ffff	0	TODO
VC0_RX_FLOW_CTRL_COMPL_DATA		Ram	000-fff	0	TODO
VC0_RX_FLOW_CTRL_COMPL_HEADER_DATA		Ram	00-ff	0	TODO

continues on next page

Table 9 – continued from previous page

Name	Instance	Type	Values	Default	Documentation
VC0_RX_FLOW_CTRL_NONPOSTED_DATA_DATA		Bool	00-ff	0	TODO
VC0_RX_FLOW_CTRL_NONPOSTED_HEADER_DATA		Bool	00-ff	0	TODO
VC0_RX_FLOW_CTRL_POSTED_DATA_DATA		Bool	000-fff	0	TODO
VC0_RX_FLOW_CTRL_POSTED_HEADER_DATA		Bool	00-ff	0	TODO
VC1_CLK_ENABLE		Bool	t/f	f	TODO
VC_ENABLE		Bool	t/f	f	TODO
VSEC_CAP_DATA		Ram	0-f	0	TODO
VSEC_ID_DATA		Ram	0000-ffff	0	TODO
ASPM_OPTIONALITY		Bool	t/f	f	TODO
BAR0_64BIT_MEMSPACE		Bool	t/f	f	TODO
BAR0_IO_SPACE 0-7		Bool	t/f	f	TODO
BAR0_PREFETCHABLE		Bool	t/f	f	TODO
BAR0_SIZE_MASK 7DATA		Ram	28 bits	0	TODO
BAR1_64BIT_MEMSPACE		Mux	<ul style="list-style-type: none"> disabled enabled all_one 	disabled	TODO
BAR1_IO_SPACE 0-7		Bool	t/f	f	TODO
BAR1_PREFETCHABLE		Bool	t/f	f	TODO
BAR1_SIZE_MASK 7DATA		Ram	28 bits	0	TODO
BAR2_64BIT_MEMSPACE		Bool	t/f	f	TODO
BAR2_IO_SPACE 0-7		Bool	t/f	f	TODO
BAR2_PREFETCHABLE		Bool	t/f	f	TODO
BAR2_SIZE_MASK 7DATA		Ram	28 bits	0	TODO
BAR3_64BIT_MEMSPACE		Mux	<ul style="list-style-type: none"> disabled enabled all_one 	disabled	TODO
BAR3_IO_SPACE 0-7		Bool	t/f	f	TODO
BAR3_PREFETCHABLE		Bool	t/f	f	TODO
BAR3_SIZE_MASK 7DATA		Ram	28 bits	0	TODO
BAR4_64BIT_MEMSPACE		Bool	t/f	f	TODO
BAR4_IO_SPACE 0-7		Bool	t/f	f	TODO
BAR4_PREFETCHABLE		Bool	t/f	f	TODO
BAR4_SIZE_MASK 7DATA		Ram	28 bits	0	TODO
BAR5_64BIT_MEMSPACE		Mux	<ul style="list-style-type: none"> disabled enabled all_one 	disabled	TODO
BAR5_IO_SPACE 0-7		Bool	t/f	f	TODO
BAR5_PREFETCHABLE		Bool	t/f	f	TODO
BAR5_SIZE_MASK 7DATA		Ram	28 bits	0	TODO
BRIDGE_PORT_SSID_SUPPORT		Bool	t/f	f	TODO
BRIDGE_PORT_VGA_ENABLE		Bool	t/f	f	TODO
CLASS_CODE_DATA		Ram	24 bits	0	TODO

continues on next page

Table 9 – continued from previous page

Name	Instance	Type	Values	Default	Documentation
COMPLETION_TIMEOUT		Mux	<ul style="list-style-type: none"> • cmpl_a • cmpl_ab • cmpl_abc • cmpl_abcd • cmpl_b • cmpl_bc • cmpl_bcd • disabled 	cmpl_a	TODO
D0_PME	0-7	Bool	t/f	f	TODO
D1_PME	0-7	Bool	t/f	f	TODO
D1_SUPPORT	0-7	Bool	t/f	f	TODO
D2_PME	0-7	Bool	t/f	f	TODO
D2_SUPPORT	0-7	Bool	t/f	f	TODO
D3_COLD_PME	0-7	Bool	t/f	f	TODO
D3_HOT_PME	0-7	Bool	t/f	f	TODO
DEEMPHASIS_ENABLE		Bool	t/f	f	TODO
DEVICE_ID_DATA	0-7	Ram	0000-ffff	0	TODO
DEVICE_SPECIFIC_INIT		Bool	t/f	f	TODO
DIFFCLOCK_NFTS_COUNT_DATA		Ram	00-ff	0	TODO
DISABLE_SNOOP_PACKET		Bool	t/f	f	TODO
DLL_ACTIVE_REPORT_SUPPORT		Bool	t/f	f	TODO
ECRC_CHECK_CAPABLE		Bool	t/f	f	TODO
ECRC_GEN_CAPABLE		Bool	t/f	f	TODO
EIE_BEFORE_NFTS_COUNT_DATA		Ram	0-f	0	TODO
ELECTROMECH_INTERLOCK		Bool	t/f	f	TODO
ENABLE_COMPLETION_TIMEOUT_DISABLE		Bool	t/f	f	TODO
ENABLE_FUNCTION_MSIX_SUPPORT		Bool	t/f	f	TODO
ENABLE_L0S_ASPM		Bool	t/f	f	TODO
ENABLE_L1_ASPM		Bool	t/f	f	TODO
ENDPOINT_L0_LATENCY_DATA		Ram	0-7	0	TODO
ENDPOINT_L1_LATENCY_DATA		Ram	0-7	0	TODO
EXPANSION_BASE7ADDRESS_REGISTER_DATA_0		32 bits		0	TODO
EXTEND_TAG_FIELD		Bool	t/f	f	TODO
FLR_CAPABILITY	0-7	Bool	t/f	f	TODO
GEN2_DIFFCLOCK_NFTS_COUNT_DATA		Ram	00-ff	0	TODO
GEN2_SAMECLOCK_NFTS_COUNT_DATA		Ram	00-ff	0	TODO
HOT_PLUG_SUPPORT_DATA		Ram	00-7f	0	TODO
INDICATOR_DATA	0-7	Ram	0-7	0	TODO
INTEL_ID_ACCESS	357	Bool	t/f	f	TODO
INTERRUPT_PIN	0-7	Mux	<ul style="list-style-type: none"> • disabled • inta • intb • intc • intd 	disabled	TODO

continues on next page

Table 9 – continued from previous page

Name	Instance	Type	Values	Default	Documentation
IO_WINDOW_ADDDR_WIDTH		Mux	<ul style="list-style-type: none"> disabled window_16_bit window_32_bit 	disabled	TODO
L0_EXIT_LATENCY	DIFFCLOCK_DATA	DATA	0-7	0	TODO
L0_EXIT_LATENCY	SAMECLOCK_DATA	DATA	0-7	0	TODO
L1_EXIT_LATENCY	DIFFCLOCK_DATA	DATA	0-7	0	TODO
L1_EXIT_LATENCY	SAMECLOCK_DATA	DATA	0-7	0	TODO
L2_ASYNC_LOGIC	0-7	Bool	t/f	f	TODO
LOW_PRIORITY_NC		Bool	t/f	f	TODO
MAXIMUM_CURRENT_DATA		Ram	0-7	0	TODO
MAX_LINK_WIDTH	0-7	Mux	<ul style="list-style-type: none"> disabled x4 x2 x1 x8 	disabled	TODO
MAX_PAYLOAD_SIZE		Num	<ul style="list-style-type: none"> 128 256 512 	128	TODO
MSIX_PBA_BIR	DATA	Ram	0-7	0	TODO
MSIX_PBA_OFFSET	DATA	Ram	29 bits	0	TODO
MSIX_TABLE_BIR	DATA	Ram	0-7	0	TODO
MSIX_TABLE_OFFSET	DATA	Ram	29 bits	0	TODO
MSIX_TABLE_SIZE	DATA	Ram	000-7ff	0	TODO
MSI_64BIT_ADDRESSING_CAPABLE		Bool	t/f	f	TODO
MSI_MASKING_CAPABLE		Bool	t/f	f	TODO
MSI_MULTI_MESSAGE_CAPABLE		Num	<ul style="list-style-type: none"> 1-2 4 8 16 32 	1	TODO
MSI_SUPPORT	0-7	Bool	t/f	f	TODO
NO_COMMAND_COMPLETED		Bool	t/f	f	TODO
NO_SOFT_RESET	0-7	Bool	t/f	f	TODO
PCIE_SPEC_VERSION	0-7	Num	<ul style="list-style-type: none"> 0-2 	0	TODO

continues on next page

Table 9 – continued from previous page

Name	Instance	Type	Values	Default	Documentation
PORTTYPE_FUNC0-7		Mux	<ul style="list-style-type: none"> • ep_native • ep_legacy • rp • sw_up • sw_dn • bridge • switch_mode • shared_mode 	ep_native	TODO
PREFETCHABLE_MEM_WINDOW_ADDR_WIDTH			<ul style="list-style-type: none"> • 0 • 32 • 64 	0	TODO
REVISION_ID_DATA		Ram	00-ff	0	TODO
ROLE_BASED_ERROR_REPORTING		Bool	t/f	f	TODO
RX_EI_L0S	0-7	Bool	t/f	f	TODO
SAMECLOCK_NPTS_COUNT_DATA		Ram	00-ff	0	TODO
SLOT_NUMBER_DATA		Ram	0000-1fff	0	TODO
SLOT_POWER_LIMIT_DATA		Ram	00-ff	0	TODO
SLOT_POWER_SCALE_DATA		Ram	0-3	0	TODO
SSID_DATA	0-7	Ram	0000-ffff	0	TODO
SSVID_DATA	0-7	Ram	0000-ffff	0	TODO
SUBSYSTEM_DEVICE_ID_DATA_0		Ram	0000-ffff	0	TODO
SUBSYSTEM_VENDOR_ID_DATA_0		Ram	0000-ffff	0	TODO
SURPRISE_DOWN_ERROR_SUPPORT		Bool	t/f	f	TODO
USE_AER	0-7	Bool	t/f	f	TODO
VC_ARBITRATION_0-7		Bool	t/f	f	TODO
VENDOR_ID_DATA_7		Ram	0000-ffff	0	TODO
ALTPE2_HIP_BASE5ADDR_USER		Ram	000-3ff	0	TODO
CVP_MDIO_DIS_CSR_CTRL_1		Bool	t/f	f	TODO
DFT_BROADCAST_EN_1		Bool	t/f	f	TODO
FORCE_MDIO_DIS_CSR_CTRL_1		Bool	t/f	f	TODO
POWER_ISOLATION_EN_1_DATA		Bool	t/f	f	TODO

2.3.13 DLL

The Delay-Locked loop does phase control for the DQS16.

TODO: everything

Name	Type	Values	Default	Documentation
A5_COUNTER_INIT	Num	<ul style="list-style-type: none"> • 3 • 12 • 24 • 40 • 48 • 72 • 80 • 96 	3	TODO
ALOAD_INVERT_EN	Bool	t/f	f	TODO
ARMSTRONG_EN	Bool	t/f	f	TODO
DELAY_CHAIN_GLITCH_CTRL_EN	Bool	t/f	f	TODO
DELAY_CONTROL	Mux	<ul style="list-style-type: none"> • bit7 • static 	static	TODO
DLL_ADDI_EN	Bool	t/f	f	TODO
DLL_INPUT	Mux	<ul style="list-style-type: none"> • vss • sd_pll0 • sd_pll1 • cn_pll0 • cn_pll1 • tb_pll0 • tb_pll1 	vss	TODO
DLL_RD_PD	Ram	0-7	0	TODO
JITTER_COUNTER_EN	Bool	t/f	t	TODO
JITTER_REDUCE_EN	Bool	t/f	t	TODO
RB_CO	Ram	0-3	3	TODO
STATIC_DLL_SETTING	Ram	00-7f	0	TODO
UPDNEN_EN	Bool	t/f	t	TODO
UPNDNIN	Mux	<ul style="list-style-type: none"> • bit4 • core 	core	TODO
UPNDNIN_EN	Bool	t/f	t	TODO
UPNDNIN_INVERT_EN	Bool	t/f	t	TODO
UPNDNIN_INV_EN	Bool	t/f	t	TODO
UPWNDCORE	Mux	<ul style="list-style-type: none"> • upndn • updnen • up_ndn • refclk 	upndn	TODO
USE_ALOAD	Bool	t/f	t	TODO

2.3.14 SERPAR

Unclear yet.

TODO: everything

Name	Type	Values	Default	Documentation
ENSER_SELECT	Mux	<ul style="list-style-type: none"> disabled block_0 block_1 block_2 block_3 	disabled	TODO

2.3.15 LVL

The Leveling Delay Chain does something linked to the DQS16.

TODO: everything

Name	Instance	Type	Values	Default	Documentation
ADDI_EN		Bool	t/f	f	TODO
CO_DELAY		Ram	0-3	3	TODO
DLL_SEL		Ram	0-1	0	TODO
FBOUT0_DELAY		Ram	0-3	0	TODO
FBOUT0_DELAY_PWR_SVG_EN		Bool	t/f	t	TODO
FBOUT1_DELAY		Ram	0-3	0	TODO
FBOUT1_DELAY_PWR_SVG_EN		Bool	t/f	t	TODO
PHYCLK_GATING_DIS		Bool	t/f	f	TODO
PHYCLK_SEL		Ram	0-3	0	TODO
PHYCLK_SEL_INV_EN		Bool	t/f	f	TODO
CLK_DELAY	0-3	Ram	0-3	0	TODO
CLK_DELAY_PWR_SVG_EN		Bool	t/f	f	TODO
CLK_GATING_DIS	0-3	Bool	t/f	f	TODO
CORE_INV_EN	0-3	Bool	t/f	f	TODO
DELAY_CLK_SEL	0-3	Mux	<ul style="list-style-type: none"> core pll 	core	TODO
PLL_SEL	0-3	Num	<ul style="list-style-type: none"> 1-3 	1	TODO

2.3.16 TERM

The TERM blocks control the On-Chip Termination circuitry

TODO: everything

Name	Type	Values	Default	Documentation
CALCLR_EN	Bool	t/f	f	TODO
CAL_MODE	Mux	<ul style="list-style-type: none"> disabled rs_12_15v rs_18_30v 	disabled	TODO
CLKENUSR_INV	Bool	t/f	f	TODO
ENSERUSR_INV	Bool	t/f	f	TODO
INTOSC_2_EN	Bool	t/f	t	TODO
NCLRUSR_INV	Bool	t/f	f	TODO
PLLBIAS_EN	Bool	t/f	f	TODO
POWERUP	Bool	t/f	f	TODO
RSADJUST_VAL	Mux	<ul style="list-style-type: none"> disabled rsadjust_10 rsadjust_6p5 rsadjust_3 rsadjust_m3 rsadjust_m6 rsadjust_m9 rsadjust_m12 	disabled	TODO
RSHIFT_RDOWN_DIS	Bool	t/f	f	TODO
RSHIFT_RUP_DIS	Bool	t/f	f	TODO
RSMULT_VAL	Mux	<ul style="list-style-type: none"> disabled rsmult_1 rsmult_2 rsmult_3 rsmult_4 rsmult_5 rsmult_6 rsmult_7 rsmult_10 	rsmult_1	TODO
RTADJUST_VAL	Mux	<ul style="list-style-type: none"> disabled rtadjust_2p5v rtadjust_1p5_1p8v 	disabled	TODO
RTMULT_VAL	Mux	<ul style="list-style-type: none"> disabled rtmult_1 rtmult_2 rtmult_3 rtmult_4 rtmult_5 rtmult_6 	rtmult_1	TODO
SCANEN_INV	Bool	t/f	f	TODO
TEST_0_EN	Bool	t/f	f	TODO
TEST_1_EN	Bool	t/f	f	TODO
TEST_4_EN	Bool	t/f	f	TODO
TEST_5_EN	Bool	t/f	f	TODO
USER_OCT_INV	Bool	t/f	f	TODO
VREFH_LEVEL	Mux	<ul style="list-style-type: none"> vref_m vref_l vref_h 	vref_m	TODO

2.3.17 PMA3

The PMA3 blocks control triplets of channels used with the HSSI.

TODO: everything

Name	Instance	Type	Values	Default	Documentation
FPLL_DRV_EN		Bool	t/f	t	TODO
FPLL_REFCLK_SEL_IQ_TX_RX_CLK		Mux	<ul style="list-style-type: none"> iq_tx_rx_clk0 iq_tx_rx_clk1 iq_tx_rx_clk2 iq_tx_rx_clk3 iq_tx_rx_clk4 iq_tx_rx_clk5 pd 	pd	TODO
FPLL_SEL_IQ_TX_RX_CLK		Mux	<ul style="list-style-type: none"> iq_tx_rx_clk0 iq_tx_rx_clk1 iq_tx_rx_clk2 pd 	pd	TODO
FPLL_SEL_REF_IQCLK		Mux	<ul style="list-style-type: none"> ffpll_top ref_iqclk0 ref_iqclk1 ref_iqclk2 ref_iqclk3 ffpll_bot pd 	pd	TODO
FPLL_SEL_RX_IQCLK		Mux	<ul style="list-style-type: none"> rx_iqclk0 rx_iqclk1 rx_iqclk2 rx_iqclk3 pd 	pd	TODO

continues on next page

Table 10 – continued from previous page

Name	Instance	Type	Values	Default	Documentation
HCLK_TOP_OUT_DRIVER		Mux	<ul style="list-style-type: none"> • tristate • up_en • down_en 	down_en	TODO
SEGMENTED_0	UP_MUX_SEL	Mux	<ul style="list-style-type: none"> • other_segmented • pd_1 • ch0_txpll 	ch0_txpll	TODO
X6_DRIVER_EN		Bool	t/f	f	TODO
AUTO_NEGOTIATION		Bool	t/f	f	TODO
CDR_PLL_ATB	0-2	Ram	0-f	0	TODO
CDR_PLL_BBPD_CLK0_OFFSET		Mux	<ul style="list-style-type: none"> • delta_0 • delta_1_left • delta_2_left • delta_3_left • delta_4_left • delta_5_left • delta_6_left • delta_7_left • delta_1_right • delta_2_right • delta_3_right • delta_4_right • delta_5_right • delta_6_right • delta_7_right 	delta_0	TODO

continues on next page

Table 10 – continued from previous page

Name	Instance	Type	Values	Default	Documentation
CDR_PLL_BBPD_CLK180_OFFSET		Mux	<ul style="list-style-type: none"> • delta_0 • delta_1_left • delta_2_left • delta_3_left • delta_4_left • delta_5_left • delta_6_left • delta_7_left • delta_1_right • delta_2_right • delta_3_right • delta_4_right • delta_5_right • delta_6_right • delta_7_right 	delta_0	TODO

continues on next page

Table 10 – continued from previous page

Name	Instance	Type	Values	Default	Documentation
CDR_PLL_BBPD_CLK270_OFFSET		Mux	<ul style="list-style-type: none"> • delta_0 • delta_1_left • delta_2_left • delta_3_left • delta_4_left • delta_5_left • delta_6_left • delta_7_left • delta_1_right • delta_2_right • delta_3_right • delta_4_right • delta_5_right • delta_6_right • delta_7_right 	delta_0	TODO

continues on next page

Table 10 – continued from previous page

Name	Instance	Type	Values	Default	Documentation
CDR_PLL_BBPD_CLK90_OFFSET	CLK90_OFFSET	Mux	<ul style="list-style-type: none"> delta_0 delta_1_left delta_2_left delta_3_left delta_4_left delta_5_left delta_6_left delta_7_left delta_1_right delta_2_right delta_3_right delta_4_right delta_5_right delta_6_right delta_7_right 	delta_0	TODO
CDR_PLL_BBPD_SEL	SEL	Mux	<ul style="list-style-type: none"> normal testmux 	normal	TODO
CDR_PLL_CGB_CLK_EN	CLK_EN	Bool	t/f	f	TODO
CDR_PLL_CLOCK_EN	CLOCK_EN	Bool	t/f	f	TODO
CDR_PLL_COUNTER_PD_CLK_DISABLE	COUNTER_PD_CLK_DISABLE	Bool	t/f	f	TODO
CDR_PLL_CPUMP_CURRENT_TEST	CURRENT_TEST	Mux	<ul style="list-style-type: none"> normal disable test_down test_up 	normal	TODO
CDR_PLL_CPUMP_A_BYPASS_EN	CPUMP_A_BYPASS_EN	Bool	t/f	f	TODO
CDR_PLL_DIAG_REV_LOOPBACK	REV_LOOPBACK	Bool	t/f	f	TODO
CDR_PLL_FAST_CLOCK_MODE_EN	FAST_CLOCK_MODE_EN	Bool	t/f	t	TODO

continues on next page

Table 10 – continued from previous page

Name	Instance	Type	Values	Default	Documentation
CDR_PLL_FB_SED-2		Mux	<ul style="list-style-type: none"> vco_clk external_clk 	vco_clk	TODO
CDR_PLL_FREF_DPM_DIV2_EN		Bool	t/f	f	TODO
CDR_PLL_GPON_DETECTION_EN		Bool	t/f	f	TODO
CDR_PLL_IGNORE_2PHASELOCK_EN		Bool	t/f	f	TODO
CDR_PLL_LEVSHIFT_POWER_TAP		Ram	0-3	1	TODO
CDR_PLL_L_COUNTER		Num	<ul style="list-style-type: none"> 1-2 4 8 	1	TODO
CDR_PLL_M_COUNTER		Num	<ul style="list-style-type: none"> 0 4-5 8 10 12 16 20 25 32 40 50 	20	TODO
CDR_PLL_ON	0-2	Bool	t/f	f	TODO
CDR_PLL_PCIE_FREQ_MHZ		Num	<ul style="list-style-type: none"> 100 125 	100	TODO
CDR_PLL_PD_COMP_CURRENT		Num	<ul style="list-style-type: none"> 5 10 20 30 40 	5	TODO
CDR_PLL_PD_L_COUNTER		Num	<ul style="list-style-type: none"> 1-2 4 8 	1	TODO

continues on next page

Table 10 – continued from previous page

Name	Instance	Type	Values	Default	Documentation
CDR_PLL_PFD_CURRENT	CDR_PFD_CURRENT	Num	<ul style="list-style-type: none"> • 5 • 10 • 20 • 30 • 40 • 50 • 60 • 80 • 100 • 120 	20	TODO
CDR_PLL_REF_DIV	CDR_REF_DIV	Num	<ul style="list-style-type: none"> • 1-2 • 4 • 8 	1	TODO
CDR_PLL_REGULATOR_INC_PCT	CDR_REGULATOR_INC_PCT	Mux	<ul style="list-style-type: none"> • p0 • p5 • p10 • p15 • p20 • p25 • disabled 	p5	TODO
CDR_PLL_REPLICA_BIAS_DIS	CDR_REPLICA_BIAS_DIS	Bool	t/f	f	TODO
CDR_PLL_RESERVED_LOOPBACK_EN	CDR_RESERVED_LOOPBACK_EN	Bool	t/f	f	TODO
CDR_PLL_RIPPLE_CAP_CTRL_EN	CDR_RIPPLE_CAP_CTRL_EN	Bool	t/f	f	TODO
CDR_PLL_RXPLL_PFD_BW_CTRL	CDR_RXPLL_PFD_BW_CTRL	Num	<ul style="list-style-type: none"> • 170 • 240 • 300 • 600 	300	TODO
CDR_PLL_RXPLL_PFD_BW_CTRL	CDR_RXPLL_PFD_BW_CTRL	Num	<ul style="list-style-type: none"> • 1600 • 3200 • 4800 • 6400 	3200	TODO
CDR_PLL_TXPLL_HCLK_DRIVER_EN	CDR_TXPLL_HCLK_DRIVER_EN	Bool	t/f	f	TODO
CDR_PLL_VCO_AUTO_RESET_EN	CDR_VCO_AUTO_RESET_EN	Bool	t/f	t	TODO
CDR_PLL_VCO_OVERANGE_REF	CDR_VCO_OVERANGE_REF	Ram	0-3	2	TODO
CDR_PLL_VLOCK_MONITOR	CDR_VLOCK_MONITOR	Mux	<ul style="list-style-type: none"> • mon_clk • mon_data 	mon_clk	TODO
CVP_EN	0-2	Bool	t/f	f	TODO
DPRIO_REG_PLD0_DMA_IF_BADDR	DPRIO_REG_PLD0_DMA_IF_BADDR	Ram	000-7ff		TODO

continues on next page

Table 10 – continued from previous page

Name	Instance	Type	Values	Default	Documentation
FORCE_MDIO_DISABLE_CSR_END	DIS_CSR_END	Bool	t/f	f	TODO
HCLK_PCS_DRIVER_EN	DRIVER_EN	Bool	t/f	f	TODO
INT_EARLY_EIO3_SEL	EIO3_SEL	Mux	<ul style="list-style-type: none"> • pcs • core 	pcs	TODO
INT_FFCLK_EN	0-2	Bool	t/f	f	TODO
INT_LTR_SEL	0-2	Mux	<ul style="list-style-type: none"> • pcs • core 	pcs	TODO
INT_PCIE_SWITCH02_SEL	02_SEL	Mux	<ul style="list-style-type: none"> • pcs • core 	pcs	TODO
INT_TXDERECTRX2_SEL	TX2_SEL	Mux	<ul style="list-style-type: none"> • pcs • core 	pcs	TODO
INT_TX_ELEC_IDLE_SEL	IDLE_SEL	Mux	<ul style="list-style-type: none"> • pcs • core 	pcs	TODO
IQ_CLK_TO_CH20_SEL	CH20_SEL	Mux	<ul style="list-style-type: none"> • ffpll_top • ffpll_bot • ref_clk0 • ref_clk1 • ref_clk2 • ref_clk3 • rx_clk0 • rx_clk1 • rx_clk2 • rx_clk3 • pd_pma 	pd_pma	TODO

continues on next page

Table 10 – continued from previous page

Name	Instance	Type	Values	Default	Documentation
IQ_TX_RX_CLK_AB_SEL	0-2	Mux	<ul style="list-style-type: none"> • a_pma_rx_b_pma_rx • a_pcs_rx_b_pcs_rx • a_pma_tx_b_pma_rx • a_pcs_tx_b_pcs_tx • a_tri_b_pcs_rx • a_tri_b_pcs_tx • a_pcs_tx_b_tri • tristate 	tristate	TODO
IQ_TX_RX_TO_CLK_FB	0-2	Mux	<ul style="list-style-type: none"> • clk0 • clk1 • clk2 • pd 	pd	TODO
PCLK0_SEL	0-2	Ram	0-7	0	TODO
PCLK1_SEL	0-2	Ram	0-7	0	TODO
PCLK_SEL	0-2	Mux	<ul style="list-style-type: none"> • a_pma_rx_b_pma_rx • a_pcs_rx_b_pcs_rx • a_pma_tx_b_pma_rx • a_pcs_tx_b_pcs_tx • a_tri_b_pcs_rx • a_tri_b_pcs_tx • a_pcs_tx_b_tri • tristate 	tristate	TODO
RX_BIT_SLIP_BYPASS_EN	0-1	Bool	t/f	t	TODO
RX_BUF_RX_ATB	0-2	Ram	0-f	0	TODO
RX_BUF_SD_3DBW_GAIN_EN	0-1	Bool	t/f	f	TODO
RX_BUF_SD_CDCLK_TO_CGB_EN	0-1	Bool	t/f	f	TODO
RX_BUF_SD_DIAG2LOOPBACK	0-1	Bool	t/f	f	TODO
RX_BUF_SD_EN	0-2	Bool	t/f	f	TODO
RX_BUF_SD_HAIF2BW_EN	0-1	Bool	t/f	f	TODO

continues on next page

Table 10 – continued from previous page

Name	Instance	Type	Values	Default	Documentation
RX_BUF_SD_OFF0-2		Mux	<ul style="list-style-type: none"> • divrx_1 • divrx_2 • divrx_3 • divrx_4 • divrx_5 • divrx_6 • divrx_7 • divrx_8 • divrx_9 • divrx_10 • divrx_11 • divrx_12 • divrx_13 • divrx_14 • reserved_off_1 • reserved_off_2 • off_on_tx_divrx_1 • off_on_tx_divrx_2 • off_on_tx_divrx_3 • off_on_tx_divrx_4 • off_on_tx_divrx_5 • off_on_tx_divrx_6 • off_on_tx_divrx_7 • off_on_tx_divrx_8 • off_on_tx_divrx_9 • off_on_tx_divrx_10 • off_on_tx_divrx_11 • off_on_tx_divrx_12 • off_on_tx_divrx_13 • off_on_tx_divrx_14 	divrx_2	TODO

continues on next page

Table 10 – continued from previous page

Name	Instance	Type	Values	Default	Documentation
RX_BUF_SD_ON	0-2	Mux	<ul style="list-style-type: none"> • pulse_4 • pulse_6 • pulse_8 • pulse_10 • pulse_12 • pulse_14 • pulse_16 • pulse_18 • pulse_20 • pulse_22 • pulse_24 • pulse_26 • pulse_28 • pulse_30 • reserved_on_1 • reserved_on_2 • force_on 	pulse_6	TODO
RX_BUF_SD_RX_GAIN_A	0	Mux	<ul style="list-style-type: none"> • v0 • v0p5 • v0p75 • v1 	v0	TODO
RX_BUF_SD_RX_GAIN_V	0	Mux	<ul style="list-style-type: none"> • v0 • v0p5 • v0p75 • v1 	v1	TODO
RX_BUF_SD_RX_CLK_DIV2_EN	0	Bool	t/f	f	TODO
RX_BUF_SD_RX_REFCLK_EN	0	Bool	t/f	f	TODO
RX_BUF_SD_TERM2_SEL	0	Mux	<ul style="list-style-type: none"> • external • r150ohm • r120ohm • r100ohm • r85ohm 	r100ohm	TODO

continues on next page

Table 10 – continued from previous page

Name	Instance	Type	Values	Default	Documentation
RX_BUF_SD_THRESHOLD_MV		Num	<ul style="list-style-type: none"> • 15 • 20 • 25 • 30 • 35 • 40 • 45 • 50 	30	TODO
RX_BUF_SD_VCM_SEL		Mux	<ul style="list-style-type: none"> • tristated1 • tristated2 • tristated3 • tristated4 • v0p35 • v0p50 • v0p55 • v0p60 • v0p65 • v0p70 • v0p75 • v0p80 • pull_down_strong • pull_down_weak • pull_up_strong • pull_up_weak 	v0p80	TODO
RX_BUF_SX_PDB_EN		Bool	t/f	f	TODO
RX_BUF_VCM_CURRENT_ADD		Ram	0-3	1	TODO
RX_DESER_CLK_SEL		Mux	<ul style="list-style-type: none"> • or_cal • lc • pld 	or_cal	TODO
RX_DESER_REVERSE_LOOPBACK		Mux	<ul style="list-style-type: none"> • rx • cdr 	rx	TODO
RX_EN	0-2	Bool	t/f	f	TODO
RX_MODE_BITS	0-2	Num	<ul style="list-style-type: none"> • 8 • 10 • 16 • 20 	8	TODO
RX_SDCLK_EN	0-2	Bool	t/f	f	TODO

continues on next page

Table 10 – continued from previous page

Name	Instance	Type	Values	Default	Documentation
RX_VCO_BYPASS	0-2	Mux	<ul style="list-style-type: none"> • clklow • fref • normal • normal_dont_care 	normal	TODO
TX_BUF_CML_EN	0-2	Bool	t/f	f	TODO
TX_BUF_COMMON_MODE_DRIVER_SEL	0-2	Mux	<ul style="list-style-type: none"> • grounded • pull_down • pull_up • pull_up_vccela • tristated1 • tristated2 • tristated3 • tristated4 • v0p35 • v0p50 • v0p55 • v0p60 • v0p65 • v0p70 • v0p75 • v0p80 	v0p65	TODO
TX_BUF_DFT_SE0	0-2	Mux	<ul style="list-style-type: none"> • vod_en_lsb • vod_en_msb • po1_en • disabled • pre_en_po2_en 	pre_en_po2_en	TODO
TX_BUF_DRIVER_RESOLUTION_CTRL	0-2	Mux	<ul style="list-style-type: none"> • combination • disabled • offset_main • offset_po1 	offset_main	TODO
TX_BUF_EN	0-2	Bool	t/f	f	TODO

continues on next page

Table 10 – continued from previous page

Name	Instance	Type	Values	Default	Documentation
TX_BUF_FIR_COEFF_SEL		Mux	<ul style="list-style-type: none"> ram dynamic 	ram	TODO
TX_BUF_LOCAL_0H_CTL		Mux	<ul style="list-style-type: none"> r49ohm r29ohm r42ohm r22ohm 	r29ohm	TODO
TX_BUF_LST_ATTEN-2		Ram	0-f	0	TODO
TX_BUF_RX_DETECT_MODE		Ram	0-f	0	TODO
TX_BUF_RX_DETECT_PDB_EN		Bool	t/f	f	TODO
TX_BUF_SLEW_RATE_CTRL		Num	<ul style="list-style-type: none"> 15 30 50 90 160 	30	TODO
TX_BUF_SWING_BOOST_DIS		Bool	t/f	f	TODO
TX_BUF_TERM_SEL		Mux	<ul style="list-style-type: none"> r150ohm r120ohm r100ohm r85ohm external 	r100ohm	TODO
TX_BUF_VCM_CURRENT_ADD		Ram	0-3	1	TODO
TX_BUF_VOD_BOOST_DIS		Bool	t/f	f	TODO
TX_BUF_VOD_SW_2ST_POST_TAP		Ram	00-1f	0	TODO
TX_BUF_VOD_SW_MAIN_TAP		Ram	00-3f	0	TODO
TX_CGB_CLK_MODE		Mux	<ul style="list-style-type: none"> disable enable_mute enable_mute_master_channel 	disable	TODO
TX_CGB_COUNTER_RESET_EN		Bool	t/f	f	TODO
TX_CGB_ENABLE-2		Bool	t/f	f	TODO
TX_CGB_FREF_VCO_BYPASS		Bool	t/f	f	TODO
TX_CGB_MUX_POWER_DOWN		Bool	t/f	f	TODO
TX_CGB_PCIE_RESET		Mux	<ul style="list-style-type: none"> normal pcie 	normal	TODO

continues on next page

Table 10 – continued from previous page

Name	Instance	Type	Values	Default	Documentation
TX_CGB_RX_IQ0	CLK_SEL	Mux	<ul style="list-style-type: none"> cgb_x1_m_div rx_output tristate 	tristate	TODO
TX_CGB_SYNC	0-2	Mux	<ul style="list-style-type: none"> normal sync_rst 	sync_rst	TODO
TX_CGB_X1_CLK0	CLK_SOURCE_SEL	Mux	<ul style="list-style-type: none"> up_segmented down_segmented ffpll ch1_txpll_t ch2_txpll_b same_ch_txpll hfclk_xn_up hfclk_cn1_x6_dn hfclk_xn_dn hfclk_ch1_x6_up 	up_segmented	TODO
TX_CGB_X1_DIV0	MM_SEL	Num	<ul style="list-style-type: none"> 1-2 4 8 	1	TODO
TX_CGB_XN_CLK0	CLK_SOURCE_SEL	Mux	<ul style="list-style-type: none"> xn_up ch1_x6_dn xn_dn ch1_x6_up cgb_x1_m_div 	cgb_x1_m_div	TODO

continues on next page

Table 10 – continued from previous page

Name	Instance	Type	Values	Default	Documentation
TX_MODE_BITS	0-2	Num	<ul style="list-style-type: none"> • 8 • 10 • 16 • 20 • 80 	8	TODO
TX_SER_CLK_DIV2	TX_DESKEW	Ram	0-f	0	TODO
TX_SER_DUTY_CYCLE	TX_CYCLE_TIME	Ram	0-7	3	TODO
TX_SER_FORCED_DATA_MODE	TX_DATA_MODE_EN	Bool	t/f	f	TODO
TX_SER_POST_TAP2	TX_TAP2_1_EN	Bool	t/f	f	TODO
TX_VREF_ES_TAP2	0-2	Mux	<ul style="list-style-type: none"> • vref_10r_ov_18r • vref_11r_ov_19r • vref_12r_ov_20r • vref_13r_ov_21r • vref_14r_ov_22r 	vref_12r_ov_20r	TODO
REF_IQCLK_BUF0	REF_IQCLK_BUF0_EN	Bool	t/f	f	TODO
RX_IQCLK_BUF0	RX_IQCLK_BUF0_EN	Bool	t/f	f	TODO
FFPLL_IQTXRXCLK	FFPLL_IQTXRXCLK_DIRECTION	Mux	<ul style="list-style-type: none"> • tristate • up • down 	tristate	TODO
FFPLL_IQCLK_DIRECTION	FFPLL_IQCLK_DIRECTION	Mux	<ul style="list-style-type: none"> • tristate • up • down 		TODO
CLKBUF_DIV2	CLKBUF_DIV2_EN	Bool	t/f	f	TODO
CLKBUF_LVPECL	CLKBUF_LVPECL_DIS	Bool	t/f	t	TODO
CLKBUF_TERM	CLKBUF_TERM_DIS	Bool	t/f	t	TODO
CLKBUF_VCM	CLKBUF_VCM_PUP	Mux	<ul style="list-style-type: none"> • tristate • vcc 	tristate	TODO
SEGMENTED_0	SEGMENTED_0_DOWN_MUX_SEL	Mux	<ul style="list-style-type: none"> • ch2_txpll • other_segmented • pd_1 	pd_1	TODO

continues on next page

Table 10 – continued from previous page

Name	Instance	Type	Values	Default	Documentation
SEGMENTED_1_DOWN_MUX_SEL		Mux	<ul style="list-style-type: none"> • fpllin • mux1 • ch0_txpll • pd_2 	pd_2	TODO
SEGMENTED_1_UP_MUX_SEL		Mux	<ul style="list-style-type: none"> • fpllin • mux1 • ch2_txpll • pd_2 • ch1_txpll_bot • ch1_txpll_top 	ch1_txpll_top	TODO
XN_DN_SEL		Mux	<ul style="list-style-type: none"> • xn_dn • x6_up • x6_dn • pd_xn_dn 	pd_xn_dn	TODO
XN_UP_SEL		Mux	<ul style="list-style-type: none"> • xn_up • x6_up • x6_dn • pd_xn_up 	pd_xn_up	TODO
CLKBUF_DIV2_EN		Bool	t/f	f	TODO
CLKBUF_LVPECL_DIS		Bool	t/f	t	TODO
CLKBUF_TERM_DIS		Bool	t/f	t	TODO
CLKBUF_VCM_PUP		Mux	<ul style="list-style-type: none"> • tristate • vcc 	tristate	TODO
SEGMENTED_0_DOWN_MUX_SEL		Mux	<ul style="list-style-type: none"> • ch2_txpll • other_segmented • pd_1 	pd_1	TODO
SEGMENTED_1_DOWN_MUX_SEL		Mux	<ul style="list-style-type: none"> • ch1_txpll_bot • ch1_txpll_top • fpllin • mux2 • ch0_txpll • pd_2 	pd_2	TODO

continues on next page

Table 10 – continued from previous page

Name	Instance	Type	Values	Default	Documentation
SEGMENTED_1	UP_MUX_SEL	Mux	<ul style="list-style-type: none"> • fpllin • mux2 • pd_2 • ch2_txpll 	ch2_txpll	TODO

2.3.18 HMC

The Hardware memory controller controls sets of GPIOs to implement modern SDR and DDR memory interfaces. In the sx dies one of them is taken over by the HPS. They can be bypassed in favor of direct access to the GPIOs.

TODO: everything, and in particular the hmc-input -> GPIO input mapping when bypassed.

Name	Instance	Type	Values	Default	Documentation
AC_DELAY_EN		Ram	0-3	0	TODO
ADDR_ORDER		Mux	<ul style="list-style-type: none"> • chip_row_bank_col • chip_bank_row_col • row_chip_bank_col 	chip_row_bank_col	TODO
ATTR_COUNTER_ONE_MASK		Ram	64 bits	0	TODO
ATTR_COUNTER_ONE_MATCH		Ram	64 bits	0	TODO
ATTR_COUNTER_ONE_RESET		Ram	0-1	0	TODO
ATTR_COUNTER_ZERO_MASK		Ram	64 bits	0	TODO
ATTR_COUNTER_ZERO_MATCH		Ram	64 bits	0	TODO
ATTR_COUNTER_ZERO_RESET		Ram	0-1	0	TODO
ATTR_DEBUG_SELECT_BYTE		Ram	32 bits	0	TODO
ATTR_STATIC_CONFIG_VALID		Bool	t/f	f	TODO
A_CSR_ATPG_EN		Bool	t/f	f	TODO
A_CSR_LPDDR_DIS		Bool	t/f	f	TODO
A_CSR_PIPELINEGLOBALENABL		Bool	t/f	f	TODO
A_CSR_RESET_DELAY_EN		Bool	t/f	f	TODO
A_CSR_WRAP_BC_EN		Bool	t/f	f	TODO
CAL_REQ		Bool	t/f	f	TODO
CFG_BURST_LENGTH		Num	<ul style="list-style-type: none"> • 0 • 2 • 4 • 8 • 16 	0	TODO

continues on next page

Table 11 – continued from previous page

Name	Instance	Type	Values	Default	Documentation
CFG_INTERFACE_WIDTH		Num	<ul style="list-style-type: none"> • 0 • 8 • 16 • 24 • 32 • 40 	0	TODO
CFG_SELF_RFSH_EXIT_CYCLES		Num	<ul style="list-style-type: none"> • 0 • 37 • 44 • 52 • 59 • 74 • 88 • 200 • 512 	0	TODO
CFG_STARVE_LIMIT		Ram	00-3f	0	TODO
CFG_TYPE		Mux	<ul style="list-style-type: none"> • ddr • ddr2 • ddr3 • lpddr • lpddr2 	ddr	TODO
CLR_INTR		Bool	t/f	f	TODO
CTL_ECC_ENABLED		Bool	t/f	f	TODO
CTL_ECC_RMW_ENABLED		Bool	t/f	f	TODO
CTL_REGDIMM_ENABLED		Bool	t/f	f	TODO
CTL_USR_REFRESH		Bool	t/f	f	TODO
DATA_WIDTH		Num	<ul style="list-style-type: none"> • 16 • 32 • 64 	16	TODO
DBE_INTR		Bool	t/f	f	TODO
DDIO_ADDR_EN		Ram	0000-ffff	0	TODO
DDIO_BA_EN		Ram	0-7	0	TODO
DDIO_CAS_N_EN		Bool	t/f	f	TODO
DDIO_CKE_EN		Ram	0-3	0	TODO
DDIO_CS0_N_EN		Ram	0-3	0	TODO
DDIO_DM_EN		Ram	00-1f	0	TODO
DDIO_DQSB_EN		Ram	00-1f	0	TODO
DDIO_DQSLOGIC_EN		Ram	00-1f	0	TODO
DDIO_DQS_EN		Ram	00-1f	0	TODO
DDIO_DQ_EN		Ram	45 bits	0	TODO
DDIO_MEM_CLK_EN		Bool	t/f	f	TODO
DDIO_MEM_CLK_N_EN		Bool	t/f	f	TODO

continues on next page

Table 11 – continued from previous page

Name	Instance	Type	Values	Default	Documentation
DDIO_ODT_EN		Ram	0-3	0	TODO
DDIO_RAS_N_EN		Bool	t/f	f	TODO
DDIO_RESET_N_EN		Bool	t/f	f	TODO
DDIO_WE_N_EN		Bool	t/f	f	TODO
DELAY_BONDING		Ram	0-3	0	TODO
DFX_BYPASS_ENABLE		Bool	t/f	f	TODO
DISABLE_MERGING		Bool	t/f	f	TODO
DQA_DELAY_EN		Ram	0-3	0	TODO
DQSLOGIC_DELAY_EN		Ram	0-3	0	TODO
DQ_DELAY_EN		Ram	0-3	0	TODO
ENABLE_ATPG		Bool	t/f	f	TODO
ENABLE_BONDING_WRAPBACK		Bool	t/f	f	TODO
ENABLE_BURST_INTERRUPT		Bool	t/f	f	TODO
ENABLE_BURST_TERMINATE		Bool	t/f	f	TODO
ENABLE_DQS_TRACKING		Bool	t/f	f	TODO
ENABLE_ECC_CODE_OVERWRITE		Bool	t/f	f	TODO
ENABLE_INTR		Bool	t/f	f	TODO
ENABLE_NO_DM		Bool	t/f	f	TODO
ENABLE_PIPELINEGLOBAL		Bool	t/f	f	TODO
EXTRA_CTL_CLK_ACT_TO_ACT		Ram	0-f	0	TODO
EXTRA_CTL_CLK_ACT_TO_ACT_DIFF_BANK		Bool	0-f	0	TODO
EXTRA_CTL_CLK_ACT_TO_PCH		Ram	0-f	0	TODO
EXTRA_CTL_CLK_ACT_TO_RDWR		Ram	0-f	0	TODO
EXTRA_CTL_CLK_ARF_PERIOD		Ram	0-f	0	TODO
EXTRA_CTL_CLK_ARF_TO_VALID		Ram	0-f	0	TODO
EXTRA_CTL_CLK_FOUR_ACT_TO_ACT		Bool	0-f	0	TODO
EXTRA_CTL_CLK_PCH_ALL_TO_VALID		Bool	0-f	0	TODO
EXTRA_CTL_CLK_PCH_TO_VALID		Ram	0-f	0	TODO
EXTRA_CTL_CLK_PDN_PERIOD		Ram	0-f	0	TODO
EXTRA_CTL_CLK_PDN_TO_VALID		Ram	0-f	0	TODO
EXTRA_CTL_CLK_RD_AP_TO_VALID		Ram	0-f	0	TODO
EXTRA_CTL_CLK_RD_TO_PCH		Ram	0-f	0	TODO
EXTRA_CTL_CLK_RD_TO_RD		Ram	0-f	0	TODO
EXTRA_CTL_CLK_RD_TO_RD_DIFF_CHIP		Bool	0-f	0	TODO
EXTRA_CTL_CLK_RD_TO_WR		Ram	0-f	0	TODO
EXTRA_CTL_CLK_RD_TO_WR_BANK		Ram	0-f	0	TODO
EXTRA_CTL_CLK_RD_TO_WR_DIFF_CHIP		Bool	0-f	0	TODO
EXTRA_CTL_CLK_SRF_TO_VALID		Ram	0-f	0	TODO
EXTRA_CTL_CLK_SRF_TO_ZQ_CAL		Ram	0-f	0	TODO
EXTRA_CTL_CLK_WR_AP_TO_VALID		Bool	0-f	0	TODO
EXTRA_CTL_CLK_WR_TO_PCH		Ram	0-f	0	TODO
EXTRA_CTL_CLK_WR_TO_RD		Ram	0-f	0	TODO
EXTRA_CTL_CLK_WR_TO_RD_BANK		Ram	0-f	0	TODO
EXTRA_CTL_CLK_WR_TO_RD_DIFF_CHIP		Bool	0-f	0	TODO
EXTRA_CTL_CLK_WR_TO_WR		Ram	0-f	0	TODO
EXTRA_CTL_CLK_WR_TO_WR_DIFF_CHIP		Bool	0-f	0	TODO
GANGED_ARF		Bool	t/f	f	TODO
GEN_DBE		Ram	0-1	0	TODO
GEN_SBE		Ram	0-1	0	TODO

continues on next page

Table 11 – continued from previous page

Name	Instance	Type	Values	Default	Documentation
IF_DQS_WIDTH		Num	<ul style="list-style-type: none"> • 0-5 	0	TODO
INC_SYNC		Num	<ul style="list-style-type: none"> • 2-3 	2	TODO
LOCAL_IF_CS_WIDTH		Num	<ul style="list-style-type: none"> • 0-4 	0	TODO
MASK_CORR_DROPPED_INTR		Bool	t/f	f	TODO
MEM_AUTO_PD_CYCLES		Ram	0000-ffff	0	TODO
MEM_CLK_ENTRY_CYCLES		Ram	0-f	0	TODO
MEM_IF_AL		Num	<ul style="list-style-type: none"> • 0-10 	0	TODO
MEM_IF_BANKADDR_WIDTH		Num	<ul style="list-style-type: none"> • 0 • 2-3 	0	TODO
MEM_IF_COLADDR_WIDTH		Num	<ul style="list-style-type: none"> • 0 • 8-12 	0	TODO
MEM_IF_ROWADDR_WIDTH		Num	<ul style="list-style-type: none"> • 0 • 12-16 	0	TODO
MEM_IF_TCCD		Num	<ul style="list-style-type: none"> • 0-4 	0	TODO
MEM_IF_TCL		Num	<ul style="list-style-type: none"> • 0 • 3-11 	0	TODO
MEM_IF_TCWL		Num	<ul style="list-style-type: none"> • 0-8 	0	TODO
MEM_IF_TFAW		Num	<ul style="list-style-type: none"> • 0-32 	0	TODO
MEM_IF_TMRD		Num	<ul style="list-style-type: none"> • 0 • 2 • 4 	0	TODO
MEM_IF_TRAS		Num	<ul style="list-style-type: none"> • 0-29 	0	TODO

continues on next page

Table 11 – continued from previous page

Name	Instance	Type	Values	Default	Documentation
MEM_IF_TRC		Num	<ul style="list-style-type: none"> • 0-40 	0	TODO
MEM_IF_TRCD		Num	<ul style="list-style-type: none"> • 0-11 	0	TODO
MEM_IF_TREFI		Ram	0000-1fff	0	TODO
MEM_IF_TRFC		Ram	00-ff	0	TODO
MEM_IF_TRP		Num	<ul style="list-style-type: none"> • 0 • 2-10 	0	TODO
MEM_IF_TRRD		Num	<ul style="list-style-type: none"> • 0-6 	0	TODO
MEM_IF_TRTP		Num	<ul style="list-style-type: none"> • 0-8 	0	TODO
MEM_IF_TWR		Num	<ul style="list-style-type: none"> • 0-12 	0	TODO
MEM_IF_TWTR		Num	<ul style="list-style-type: none"> • 0-6 	0	TODO
MMR_CFG_MEM_BL		Num	<ul style="list-style-type: none"> • 2 • 4 • 8 • 16 	2	TODO
OUTPUT_REGD		Bool	t/f	f	TODO
PDN_EXIT_CYCLES		Mux	<ul style="list-style-type: none"> • disabled • fast • slow 	disabled	TODO
POWER_SAVING_EXIT_CYCLES		Ram	0-f	0	TODO
PRIORITY_REMAP		Mux	<ul style="list-style-type: none"> • disabled • priority_0 • priority_1 • priority_2 • priority_3 • priority_4 • priority_5 • priority_6 • priority_7 	disabled	TODO

continues on next page

Table 11 – continued from previous page

Name	Instance	Type	Values	Default	Documentation
READ_ODT_CHIP		Mux	<ul style="list-style-type: none"> disabled read_chip0_odt0_chip1 read_chip0_odt1_chip1 read_chip0_odt01_chip1 read_chip0_chip1_odt0 read_chip0_odt0_chip1_odt0 read_chip0_odt1_chip1_odt0 read_chip0_odt01_chip1_odt0 read_chip0_chip1_odt1 read_chip0_odt0_chip1_odt1 read_chip0_odt1_chip1_odt1 read_chip0_odt01_chip1_odt1 read_chip0_chip1_odt01 read_chip0_odt0_chip1_odt01 read_chip0_odt1_chip1_odt01 read_chip0_odt01_chip1_odt01 	disabled	TODO
REORDER_DATA		Bool	t/f	f	TODO
SBE_INTR		Bool	t/f	f	TODO
TEST_MODE		Bool	t/f	f	TODO
USER_ECC_EN		Bool	t/f	f	TODO

continues on next page

Table 11 – continued from previous page

Name	Instance	Type	Values	Default	Documentation
WRITE_ODT_CHIP		Mux	<ul style="list-style-type: none"> disabled write_chip0_odt0_chip1 write_chip0_odt1_chip1 write_chip0_odt01_chip1 write_chip0_chip1_odt0 write_chip0_odt0_chip1_odt0 write_chip0_odt1_chip1_odt0 write_chip0_odt01_chip1_odt0 write_chip0_chip1_odt1 write_chip0_odt0_chip1_odt1 write_chip0_odt1_chip1_odt1 write_chip0_odt01_chip1_odt1 write_chip0_chip1_odt01 write_chip0_odt0_chip1_odt01 write_chip0_odt1_chip1_odt01 write_chip0_odt01_chip1_odt01 	disabled	TODO
INST_ROM_DATA	A0-127	Ram	20 bits	0	TODO
AC_ROM_DATA	0-39	Ram	30 bits	0	TODO
AUTO_PCH_ENABLE	0-1	Bool	t/f	f	TODO
CLOCK_OFF	0-5	Bool	t/f	f	TODO
CPORT_RDY_ALMOST_FULL	0-1	Bool	t/f	f	TODO
CPORT_RFIFO_MAP	0-3	Ram	0-3	0	TODO
CPORT_TYPE	0-5	Mux	<ul style="list-style-type: none"> disabled write read bi_direction 	disabled	TODO
CPORT_WFIFO_MAP	0-3	Ram	0-3	0	TODO
CYC_TO_RLD_JARS	0-5	Ram	00-ff	0	TODO
ENABLE_BONDING	0-1	Bool	t/f	f	TODO

continues on next page

Table 11 – continued from previous page

Name	Instance	Type	Values	Default	Documentation
PORT_WIDTH	0-5	Num	<ul style="list-style-type: none"> • 32 • 64 • 128 • 256 	32	TODO
RCFG_STATIC_WEIGHT	0-3	Ram	00-1f	0	TODO
RCFG_USER_PRIORITY	0-3	Ram	0-7	0	TODO
THLD_JAR1	0-5	Ram	00-3f	0	TODO
THLD_JAR2	0-5	Ram	00-3f	0	TODO
RFIFO_CPORT_MAP	0-5	Num	<ul style="list-style-type: none"> • 0-5 	0	TODO
SINGLE_READY	0-3	Mux	<ul style="list-style-type: none"> • concatenate • separate 	concatenate	TODO
SYNC_MODE	0-3	Mux	<ul style="list-style-type: none"> • asynchronous • synchronous 	asynchronous	TODO
USE_ALMOST_EMPTY	0-1	Bool	t/f	f	TODO
WFIFO_CPORT_MAP	0-5	Num	<ul style="list-style-type: none"> • 0-5 	0	TODO
WFIFO_RDY_ALMOST_FULL	0-1	Bool	t/f	f	TODO
RCFG_SUM_WEIGHT_PRIORITY	0-3	Ram	00-ff	0	TODO

2.3.19 HPS

The interface between the FPGA and the Hard processor system is done through 37 specialized blocks or 28 different types.

TODO: everything. GOUT/GIN/DCMUX mapping is done except for HPS_CLOCKS.

HPS_BOOT

HPS_CLOCKS

Name	Instance	Type	Values	Default	Documentation
RIGHT_CLOCK_SEL	0-8	Ram	0-3	3	TODO
TOP_CLOCK_SEL	0-8	Ram	0-3	3	TODO

HPS_CLOCKS_RESETS

HPS_CROSS_TRIGGER

HPS_DBG_APB

HPS_DMA

HPS_FPGA2HPS

HPS_FPGA2SDRAM

HPS_HPS2FPGA

HPS_HPS2FPGA_LIGHT_WEIGHT

HPS_INTERRUPTS

HPS_JTAG

HPS_LOAN_IO

HPS_MPU_EVENT_STANDBY

HPS_MPU_GENERAL_PURPOSE

HPS_PERIPHERAL_CAN

(2 blocks)

HPS_PERIPHERAL_EMAC

(2 blocks)

HPS_PERIPHERAL_I2C

(4 blocks)

HPS_PERIPHERAL_NAND

HPS_PERIPHERAL_QSPI

HPS_PERIPHERAL_SDMMC

HPS_PERIPHERAL_SPI_MASTER

(2 blocks)

HPS_PERIPHERAL_SPI_SLAVE

(2 blocks)

HPS_PERIPHERAL_UART

(2 blocks)

HPS_PERIPHERAL_USB

(2 blocks)

HPS_STM_EVENT**HPS_TEST****HPS_TPIU_TRACE****2.4 Options**

Name	Type	Values	Default	Documentation
ALLOW_DEVICE_WIDE_OUTPUT_ENABLE_DIS	Bool	t/f	f	TODO
COMPRESSION_DISABLE	Bool	t/f	f	TODO
CRC_DIVIDE_ORDER	Num	<ul style="list-style-type: none"> 0-8 	0	TODO
CRC_ERROR_DETECTION_EN	Bool	t/f	f	TODO
CVPCIE_MODE	Ram	0-3	0	TODO
CVP_CONF_DONE_ENABLE	Bool	t/f	f	TODO
DEVICE_WIDE_RESET_EN	Bool	t/f	f	TODO
DRIVE_STRENGTH	Ram	0-3	0	TODO
IDCODE	Ram	00-ff		TODO
IOCSR_READY_FROM_IOCSR_DONE_EN	Bool	t/f	f	TODO
JTAG_ID	Ram	32 bits		TODO
NCEO_DIS	Bool	t/f	f	TODO
OCT_DONE_DIS	Bool	t/f	f	TODO
OPT_A	Ram	0000-ffff		TODO
OPT_B	Ram	64 bits		TODO
RELEASE_CLEAR_BEFORE_TRISTATES_LDIS	Bool	t/f	f	TODO
RETRY_CONFIG_ON_ERROR_EN	Bool	t/f	f	TODO
START_UP_CLOCK	Ram	00-ff	40	TODO

CYCLONEV LIBRARY USAGE

3.1 Library structure

The library provides a CycloneV class in the mistral namespace. Information is provided to allow to choose a CycloneV::Model object which represents a sold FPGA variant. Then a CycloneV object can be created from it. That object stores the state of the FPGA configuration and allows to read and modify it.

All the types, enums, functions, methods, arrays etc described in the following paragraph are in the CycloneV class.

3.2 Packages

```
enum package_type_t;

struct CycloneV::package_info_t {
    int pin_count;
    char type;
    int width_in_pins;
    int height_in_pins;
    int width_in_mm;
    int height_in_mm;
};

const package_info_t package_infos[5+3+3];
```

The FPGAs are sold in 11 different packages, which are named by their type (Fineline BGA, Ultra Fineline BGA or Micro Fineline BGA) and their width in mm.

Enum	Type	Pins	Size in mm	Size in pins
PKG_F17	f	256	16x16	17x17
PKG_F23	f	484	22x22	23x23
PKG_F27	f	672	26x26	27x27
PKG_F31	f	896	30x30	31x31
PKG_F35	f	1152	34x34	35x35
PKG_U15	u	324	18x18	15x15
PKG_U19	u	484	22x22	19x19
PKG_U23	u	672	28x28	23x23
PKG_M11	m	301	21x21	11x11
PKG_M13	m	383	25x25	13x13
PKG_M15	m	484	28x28	15x15

3.3 Model information

```
enum die_type_t { E50F, GX25F, GT75F, GT150F, GT300F, SX50F, SX120F };

struct Model {
    const char *name;
    const variant_info &variant;
    package_type_t package;
    char temperature;
    char speed;
    char pcie, gxb, hmc;
    uint16_t io, gpio;
};

struct variant_info {
    const char *name;
    const die_info &die;
    uint16_t idcode;
    int alut, alm, memory, dsp, dpll, dll, hps;
};

struct die_info {
    const char *name;
    die_type_t type;
    uint8_t tile_sx, tile_sy;
    // ...
};

const Model models[];
CycloneV *get_model(std::string model_name);
```

A Model is built from a package, a variant and a temperature/speed grade. A variant selects a die and which hardware is active on it.

The Model fields are:

- name - the SKU, for instance 5CSEBA6U23I7
- variant - its associated variant_info
- package - the packaging used
- temperature - the temperature grade, 'A' for automotive (-45..125C), 'I' for industrial (-40..100C), 'C' for commercial (0..85C)
- speed - the speed grade, 6-8, smaller is faster
- pcie - number of PCIe interfaces (depends on both variant and number of available pins)
- gxb - ??? (same)
- hmc - number of Memory interfaces (same)
- io - number of i/os
- gpio - number of fpga-usable gpios

The Variant fields are:

- name - name of the variant, for instance se120b
- die - its associated die_info

- `idcode` - the IDCODE associated to this variant (not unique per variant at all)
- `alut` - number of LUTs
- `alm` - number of logic elements
- `memory` - bits of memory
- `dsp` - number of dsp blocks
- `dpll` - number of pll
- `dll` - number of delay-locked loops
- `hps` - number of arm cores

The Die usable fields are:

- `name` - name of the die, for instance `sx120f`
- `type` - the enum value for the die type
- `tile_sx`, `tile_sy` - size of the tile grid

The limits indicated in the variant structure may be lower than the theoretical die capabilities. We have no idea what happens if these limits are not respected.

To create a CycloneV object, the constructor requires a Model *. Either choose one from the models array, or, in the usual case of selection by sku, the `CycloneV::get_model` function looks it up and allocates one. The models array ends with a nullptr name pointer.

The `get_model` function implements the alias “ms” for the 5CSEBA6U23I7 used in the de10-nano, a.k.a MiSTer.

3.4 pos, rnode and pnode

```
using pos_t = uint16_t;           // Tile position

static constexpr uint32_t pos2x(pos_t xy);
static constexpr uint32_t pos2y(pos_t xy);
static constexpr pos_t xy2pos(uint32_t x, uint32_t y);
```

The type `pos_t` represents a position in the grid. `xy2pos` allows to create one, `pos2x` and `pos2y` extracts the coordinates.

```
using rnode_t = uint32_t;        // Route node id

enum rnode_type_t;
const char *const rnode_type_names[];
rnode_type_t rnode_type_lookup(const std::string &n) const;

constexpr rnode_t rnode(rnode_type_t type, pos_t pos, uint32_t z);
constexpr rnode_t rnode(rnode_type_t type, uint32_t x, uint32_t y, uint32_t z);
constexpr rnode_type_t rn2t(rnode_t rn);
constexpr pos_t rn2p(rnode_t rn);
constexpr uint32_t rn2x(rnode_t rn);
constexpr uint32_t rn2y(rnode_t rn);
constexpr uint32_t rn2z(rnode_t rn);

std::string rn2s(rnode_t rn);
```

A `rnode_t` represents a node in the routing network. It is characterized by its type (`rnode_type_t`) and its coordinates (x, y for the tile, z for the instance number in the tile). Those functions allow to create one and extract the different

components. `rnode_types_names` gives the string representation for every `rnode_type_t` value, and `rnode_type_lookup` finds the `rnode_type_t` for a given name. `rn2s` provides a string representation of the `rnode` (`TYPE.xxx.yyy.zzzz`).

The `rnode_type_t` value 0 is `NONE`, and a `rnode_t` of 0 is guaranteed invalid.

```
using pnode_t = uint64_t;           // Port node id

enum block_type_t;
const char *const block_type_names[];
block_type_t block_type_lookup(const std::string &n) const;

enum port_type_t;
const char *const port_type_names[];
port_type_t port_type_lookup(const std::string &n) const;

constexpr pnode_t pnode(block_type_t bt, pos_t pos, port_type_t pt, int8_t bindex,
    ↪ int16_t pindex);
constexpr pnode_t pnode(block_type_t bt, uint32_t x, uint32_t y, port_type_t pt, int8_t
    ↪ t bindex, int16_t pindex);
constexpr block_type_t pn2bt(pnode_t pn);
constexpr port_type_t pn2pt(pnode_t pn);
constexpr pos_t pn2p(pnode_t pn);
constexpr uint32_t pn2x(pnode_t pn);
constexpr uint32_t pn2y(pnode_t pn);
constexpr int8_t pn2bi(pnode_t pn);
constexpr int16_t pn2pi(pnode_t pn);

std::string pn2s(pnode_t pn);
```

A `pnode_t` represents a port of a logical block. It is characterized by the block type (`block_type_t`), the block tile position, the block number instance (when appropriate, -1 when not), the port type (`port_type_t`) and the bit number in the port (when appropriate, -1 when not). `pn2s` provides the string representation `BLOCK.xxx.yyy(.instance):PORT(.bit)`

The `block_type_t` value 0 is `BNONE`, the `port_type_t` value 0 is `PNONE`, and `pnode_t` 0 is guaranteed invalid.

```
rnode_t pnode_to_rnode(pnode_t pn) const;
pnode_t rnode_to_pnode(rnode_t rn) const;
```

These two methods allow to find the connections between the logic block ports and the routing nodes. It is always 1:1 when there is one.

3.5 Routing network management

```
void rnode_link(rnode_t n1, rnode_t n2);
void rnode_link(pnode_t p1, rnode_t n2);
void rnode_link(rnode_t n1, pnode_t p2);
void rnode_link(pnode_t p1, pnode_t p2);
void rnode_unlink(rnode_t n2);
void rnode_unlink(pnode_t p2);
```

The method `rnode_link` links two nodes together with `n1` as source and `n2` as destination, automatically converting from `pnode_t` to `rnode_t` when needed. `rnode_unlink` disconnects anything connected to the destination `n2`.

There are two special cases. `DCMUX` is a 2:1 mux which selects between a data and a clock signal and has no disconnected state. Unlinking it puts in in the default clock position. Most `SCLK` muxes use a 5-bit vertical configuration where up to 5 inputs can be connected and the all-off configuration is not allowed. Usually at least one input goes to `vcc`, but in some cases all five are used and unlinking selects the 4th input (the default in that case).

```
std::vector<std::pair<rnode_t, rnode_t>> route_all_active_links() const;
std::vector<std::pair<rnode_t, rnode_t>> route_frontier_links() const;
```

route_all_active_links gives all current active connections. route_frontier_links solves these connections to keep only the extremities, giving the inter-logic-block connections directly.

3.6 Logic block management

```
const std::vector<pos_t> &lab_get_pos() const
[etc]
```

The numerous xxx_get_pos() methods gives the list of positions of logic blocks of a given type. The known types are lab, mlab, m10k, dsp, hps, gpio, dqs16, fpll, cmuxc, cmuxv, cmuxh, dll, hssi, cbuf, lvl, ctrl, pma3, serpar, term and hip. A vector is empty when a block type doesn't exist in the given die.

In the hps case the 37 blocks can be indexed by hps_index_t enum.

```
enum { MT_MUX, MT_NUM, MT_BOOL, MT_RAM };

enum bmux_type_t;
const char *const bmux_type_names[];
bmux_type_t bmux_type_lookup(const std::string &n) const;

struct bmux_setting_t {
    block_type_t btype;
    pos_t pos;
    bmux_type_t mux;
    int midx;
    int type;
    bool def;
    uint32_t s; // bmux_type_t, or number, or bool value, or count of bits for ram
    std::vector<uint8_t> r;
};

int bmux_type(block_type_t btype, pos_t pos, bmux_type_t mux, int midx) const;
bool bmux_get(block_type_t btype, pos_t pos, bmux_type_t mux, int midx, bmux_setting_t &s) const;
bool bmux_set(const bmux_setting_t &s);
bool bmux_m_set(block_type_t btype, pos_t pos, bmux_type_t mux, int midx, bmux_type_t &s);
bool bmux_n_set(block_type_t btype, pos_t pos, bmux_type_t mux, int midx, uint32_t s);
bool bmux_b_set(block_type_t btype, pos_t pos, bmux_type_t mux, int midx, bool s);
bool bmux_r_set(block_type_t btype, pos_t pos, bmux_type_t mux, int midx, uint64_t s);
bool bmux_r_set(block_type_t btype, pos_t pos, bmux_type_t mux, int midx, const &s);

std::vector<bmux_setting_t> bmux_get() const;
```

These methods allow to manage the logic blocks muxes configurations. A mux is characterized by its block (type and position), its type (bmux_type_t) and its instance number (0 if there is only one). There are four kinds of muxes, symbolic (MT_MUX), numeric (MT_NUM), boolean (MT_BOOL) and ram (MT_RAM).

bmux_type looks up a mux and returns its MT_* type, or -1 if it doesn't exist. bmux_get reads the state of a mux and returns it in s and true when found, false otherwise. The def field indicates whether the value is the default. The bmux_set sets a mux generically, and the bmux_*_set sets it per-type.

The no-parameter `bmux_get` version returns the state of all muxes of the FPGA.

3.7 Inverters management

```
struct inv_setting_t {
    rnode_t node;
    bool value;
    bool def;
};

std::vector<inv_setting_t> inv_get() const;
bool inv_set(rnode_t node, bool value);
```

`inv_get()` returns the state of the programmable inverters, and `inv_set` sets the state of one. The field `def` is currently very incorrect.

3.8 Pin/package management

```
enum pin_flags_t : uint32_t {
    PIN_IO_MASK      = 0x00000007,
    PIN_DPP          = 0x00000001, // Dedicated Programming Pin
    PIN_HSSI         = 0x00000002, // High Speed Serial Interface input
    PIN_JTAG         = 0x00000003, // JTAG
    PIN_GPIO         = 0x00000004, // General-Purpose I/O

    PIN_HPS          = 0x00000008, // Hardware Processor System

    PIN_DIFF_MASK    = 0x00000070,
    PIN_DM           = 0x00000010,
    PIN_DQS          = 0x00000020,
    PIN_DQS_DIS      = 0x00000030,
    PIN_DQSB         = 0x00000040,
    PIN_DQSB_DIS     = 0x00000050,

    PIN_TYPE_MASK    = 0x00000f00,
    PIN_DO_NOT_USE   = 0x00000100,
    PIN_GXP_RREF     = 0x00000200,
    PIN_NC           = 0x00000300,
    PIN_VCC          = 0x00000400,
    PIN_VCCL_SENSE   = 0x00000500,
    PIN_VCCN         = 0x00000600,
    PIN_VCCPD        = 0x00000700,
    PIN_VREF         = 0x00000800,
    PIN_VSS          = 0x00000900,
    PIN_VSS_SENSE    = 0x00000a00,
};

struct pin_info_t {
    uint8_t x;
    uint8_t y;
    uint16_t pad;
    uint32_t flags;
    const char *name;
```

(continues on next page)

(continued from previous page)

```

const char *function;
const char *io_block;
double r, c, l, length;
int delay_ps;
int index;
};

const pin_info_t *pin_find_pos(pos_t pos, int index) const;

```

The `pin_info_t` structure describes a pin with:

- x, y - its coordinates in the package grid (not the fpga grid, the pins one)
- pad - either 0xffff (no associated gpio) or (index << 14) | tile_pos, where index indicates which pad of the gpio is connected to the pin
- flags - flags describing the pin function
- name - pin name, like A1
- function - pin function as text, like “GND”
- io_block - name of the I/O block for power purposes, like 9A
- r, c, l - electrical characteristics of the pin-pad connection wire
- length - length of the wire
- delay_ps - usual signal transmission delay is ps
- index - pin sub-index for hssi_input, hssi_output, dedicated programming pins and jtag

The `pin_find_pos` method looks up a pin from a gpio tile/index combination.

3.9 Options

```

struct opt_setting_t {
    bmux_type_t mux;
    bool def;
    int type;
    uint32_t s; // bmux_type_t, or number, or bool value, or count of bits for ram
    std::vector<uint8_t> r;
};

int opt_type(bmux_type_t mux) const;
bool opt_get(bmux_type_t mux, opt_setting_t &s) const;
bool opt_set(const opt_setting_t &s);
bool opt_m_set(bmux_type_t mux, bmux_type_t s);
bool opt_n_set(bmux_type_t mux, uint32_t s);
bool opt_b_set(bmux_type_t mux, bool s);
bool opt_r_set(bmux_type_t mux, uint64_t s);
bool opt_r_set(bmux_type_t mux, const std::vector<uint8_t> &s);

std::vector<opt_setting_t> opt_get() const;

```

The options work like the block muxes without a block, tile or instance number. They’re otherwise the same.

3.10 Bitstream management

```
void clear();  
void rbf_load(const void *data, uint32_t size);  
void rbf_save(std::vector<uint8_t> &data);
```

The clear method returns the FPGA state to all defaults. rbf_load parses a raw bitstream file from memory and loads the state from it. rbf_save generates a rbf from the current state.

THE MISTRAL-CV COMMAND-LINE PROGRAM

The `mistral-cv` command line program allows for a minimal interfacing with the library. Calling it without parameters shows the possible usages.

4.1 models

```
mistral-cv models
```

Lists the known models with their SKU, IDCODE, die, variant, package, number of pins, temperature grade and speed grade.

4.2 routes

```
mistral-cv routes <model> <file.rbf>
```

Dumps the active routes in a rbf.

4.3 routes2

```
mistral-cv routes <model> <file.rbf>
```

Dumps the active routes in a rbf where a GIN/GOUT/etc does not have a port mapping associated.

4.4 cycle

```
mistral-cv cycle <model> <file.rbf> <file2.rbf>
```

Loads the rbf in `file1.rbf` and saves it back in `file2.rbf`. Useful to test if the framing/unframing of oram/pram/cram works correctly.

4.5 bels

```
mistral-cv bels <model>
```

Dumps a list of all the logic elements of a model (only depends on the die in practice).

4.6 decomp

```
mistral-cv decomp <model> <file.rbf> <file.bt>
```

Decompiles a bitstream into a compilable source. Only writes down what is identified as not being in default state.

4.7 comp

```
mistral-cv comp <file.bt> <file.rbf>
```

Compiles a source into a bitstream. The source includes the model information.

4.8 diff

```
mistral-cv diff <model> <file1.rbf> <file2.rbf>
```

Compares two rbf files and identifies the differences in terms of oram, pram and cram. Useful to list mismatches after a decomp/comp cycle.

MISTRAL CYCLONEV LIBRARY INTERNALS

5.1 Structure

A large part of the library is generated code from information in the data directory. The exception is the routing data that is converted to compressed binary and put in the gdata directory. All the conversions are done with python programs and shell scripts in the tools directory.

5.2 Routing data

The routing data is stored in bzip2-compressed text files named <die>-r.txt.bz2. Each line describes a routing mux.

A mux description looks like that:

```
H14.000.032.0003 4:0024_2832 0:GIN.000.032.0005 1:GIN.000.032.0004 2:GIN.000.032.0001
↪ 3:GIN.000.032.0000
```

That line describes the mux for the rnode H14.000.032.0003. It uses the pattern 4 as position (24, 2832) and has four inputs connected to four GIN rnodes.

The chip uses a limited number of mux types, with a specific bit pattern in the cram controlling a fixed number of inputs and of bit set/unset values selecting them. There is a total of 70 different patterns, currently only described as C++ code in cv-rpats.cc. An additional 4 are added to store the variations of pattern 6 where the default is different.

The special case of pattern 6 looks like:

```
SCLK.014.000.0025 6.3:1413_0638 0:GCLK.000.008.0009 1:RCLK.000.004.0011 4:RCLK.000.
↪ 004.0003
```

The “.3” indicates that the default is on slot 3, e.g. value 0x08 or pattern 70+3.

The python script routes-to-bin.py loads this file and generated a compressed binary version in gdata which matches the rmux structure. The script mkroutes.sh generates it for all die types.

5.3 Block muxes

The lists of block muxes and options muxes are independant of the dies. They're in the block-mux.txt files. Each mux is described in these files using the following syntax:

```
g dft_mode m:3 21.42 20.40 20.43
0 off
1 on !
7 dft_pprog
```

“g” indicates the subtype of mux, which is block-dependant, here “global”. ‘m’ indicates a symbolic mux, 3 is the number of bits. It is followed by the bits coordinates, LSB first. Here it's an inner block, so the coordinates are 2D. Options are also 2D, and peripheral blocks are 1D.

In such a case of symbolic mux it is followed by the indented possible values of the mux (in hex) with the exclamation point indicating the default.

A numeric mux is similar but the type is ‘n’ and labels on the right have to be numeric.

Boolean muxes look like this:

```
g clk0_inv          b-   6.45
```

The ‘b’ indicates boolean, and ‘-’ indicates the default is false, otherwise it is ‘+’ for true. The boolean can be multi-bits, such as in the following example. Then all bits are set or unset.

```
g pr_en             b-:2 0.61 0.67
```

Finally ram muxes look like:

```
g cvpcie_mode       r-:2   2.21 2.22
g clkin_0_src        r2:4  760 761 762 763
```

In the second case the ‘2’ between r and : indicates that the default value is 2.

Instanciated muxes can take two forms. For instance in fp1l muxes of subtype ‘c’ are instanciated on the counter number, hence have 9 values. The mux is written as:

```
c cnt_in_src          r2:2  600 601 | 602 603 | 604 605 | 606 607 | 608
↪609 | 610 611 | 612 613 | 614 615 | 616 617
c dprio0_cnt_hi_div   r1:8
* 8 9 10 11 12 13 14 15
* 24 25 26 27 28 29 30 31
* 40 41 42 43 44 45 46 47
* 56 57 58 59 60 61 62 63
* 72 73 74 75 76 77 78 79
* 88 89 90 91 92 93 94 95
* 104 105 106 107 108 109 110 111
* 120 121 122 123 124 125 126 127
* 136 137 138 139 140 141 142 143
```

Either the bits are indicated on the same line separated by ‘|’, or they're set as one set per line start with an indented ‘*’.

The lab, mlab, m10k, mlab and hps_clocks target bits in the 2D cram by offsetting from a base position computed from the tile position (see the method pos2bit). opt targets bits in the oram. All the others with the exception of pma3-c target bits in the pram from a position found in <die>-pram.txt. pma3-c targets bits in the cram from the tables in pma3-cram.txt

`mux_to_source.py` `enum <datadir>` generates the file `cv-bmuxtypes.ipp` while `mux_to_source.py mux <datadir>` generates the file `cv-bmux-data.cc`. `mkmux.sh` does both calls.

5.4 Logic blocks

Blocks come from two sources, the files `<die>-pram.txt` indicates all the peripheral blocks with their pram address. The files `<die>-<block>.txt` where `block` is `cmux`, `ctrl`, `fppl`, `hmc`, `hps` or `iob` has the information of the connections between the blocks and neighbouring blocks and the routing grid.

`blocks_to_source.py` generates the `cvd-<die>-blk.cc` file for a given die, and `mkbblocks.sh` calls it for every die.

5.5 Inverters

The list of inverters, their cram position and their default value (always 0 at this point) is in `<die>-inv.txt`. `inv_to_source.py/mkinv.sh` takes care of generating the `cvd-<die>-inv.cc` files.

5.6 Forced-1 bits

Five of the seven dies seem to have bits always set to 1. They are listed in the files `<die>-1.txt`. `blocks_to_source.py` takes care of it.

5.7 Packages

The file `<die>-pkg.txt` lists the packages and the pins of each package for each die. `pkg_to_source.py/mkpkg.sh` take cares of generating the `cvd-<die>-pkg.cc` files.

5.8 Models

`models.txt` includes all the information on variants and models. The `cv-models.cc` file is generated by `models_to_source.py` called by `mkmodels.sh`.