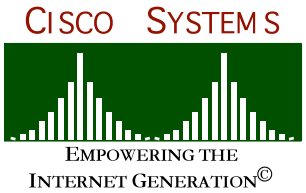


Document Number	EDCS-xxxxxx
Revision	0.1
Authors	Ken Tomei
Project Manager	Christophe Joly

R2D2 ASIC

Quick Reference Guide

Description of Purpose

This document is essentially a collection of notes gathered for personal reference. It may serve as a quick reference guide for the ASIC design team or to aid in transfer of information to cross-functional groups. For a more complete description of the ASIC and its full functionality, please refer to the Dhruva and R2D2 ASIC design specifications.

Modification History

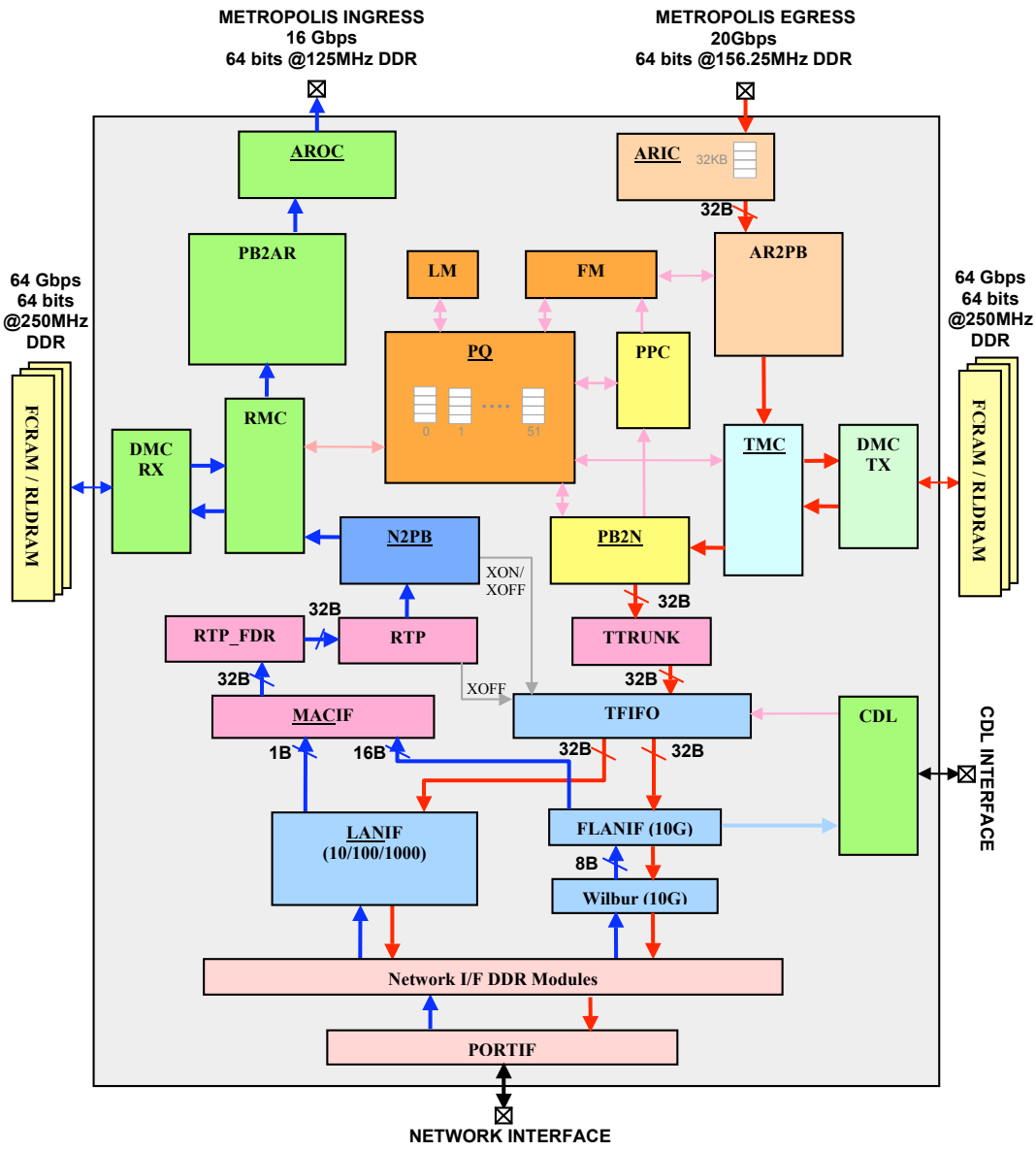
Rev.	Date	Originator	Comment
0.1	10/04/2004	Ken Tomei	Draft

Acronyms

DDR..... Double Data Rate
FCRAM..... Fast Cycle RAM
RLDRAM ..... Reduced Latency DRAM

Table of Contents

1 R2D2 Top Level Block Diagram

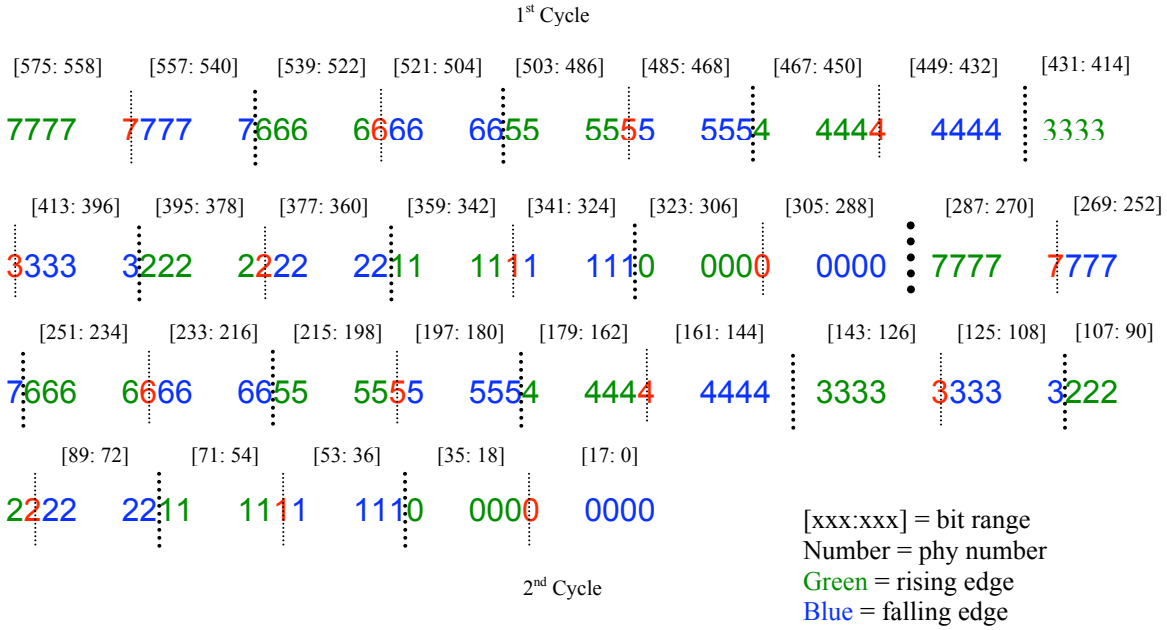


The above diagram partitions R2D2 into its logical sub-modules. Bold interconnects illustrate the main data buses, while lighter shaded lines indicate primary control communication.

Physically, the ASIC is partitioned into 8 sub-chips, whose organization is illustrated through the various module colors. The underlined portion of the module name is the sub-chip name.

2 GMTL DATA OUTPUT MAPPING

When an offline diagnostic test reports a GMTL data mismatch, it outputs the expected and received data in the format below. The following diagram may be used to help identify the location of a GMTL reported failure:



For example, bits [215:198] are connected to Phy #5, and represent the rising edge of the 2nd cycle in the burst-of-4 access.

3 RLDRAM ADDRESS MAPPING

The following section attempts to clarify how internal addresses eventually translate to the physical address pins of the RLDRAM device.

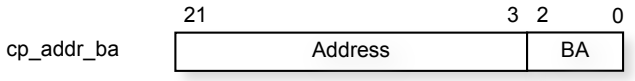
A x36 RLDRAM memory is 288Mb (32MB data + 4MB ECC). Since each write is a burst-of-4 access totaling 16B of data, there are 2 million "addresses." Thus, 21 bits are required to address the memory. For x18 devices, each write is only 8B, so 22 bits are required to fully address the memory. Since the memory is partitioned into 8 banks, 3 of those "address" bits are actually used to represent the bank address. R2D2 simply takes the internal address and uses the three least significant address bits as the bank address.

Internally, R2D2 passes around a 32-bit byte address everywhere, including within the RMC and TMC modules. The number of actual valid addresses will require fewer bits, and this is controlled through CPU programming, which is dependent on the external memory configuration of the board.

This byte address gets converted into a 22-bit cell address (1 cell = 64 bytes) before it is sent to the dual memory controller (DMC). This translation occurs in r2d_df_convert, inside r2d2_mctl (the DMC):

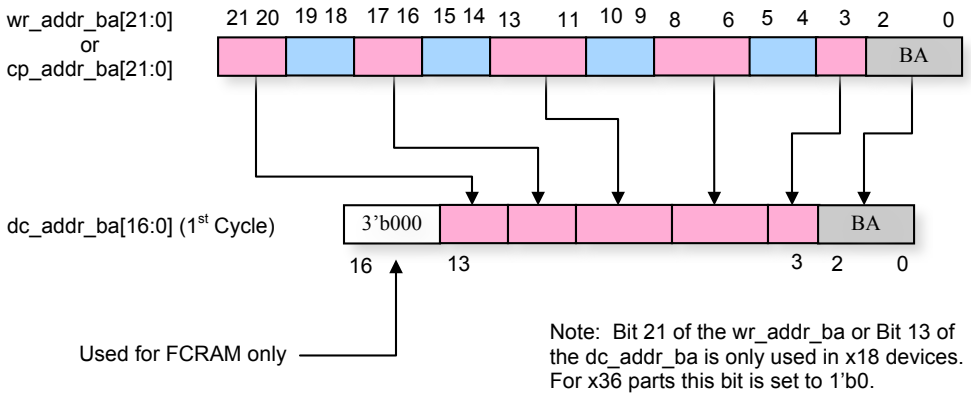
wr_addr_ba[21:0] = dh_mw_caddr[27:6]

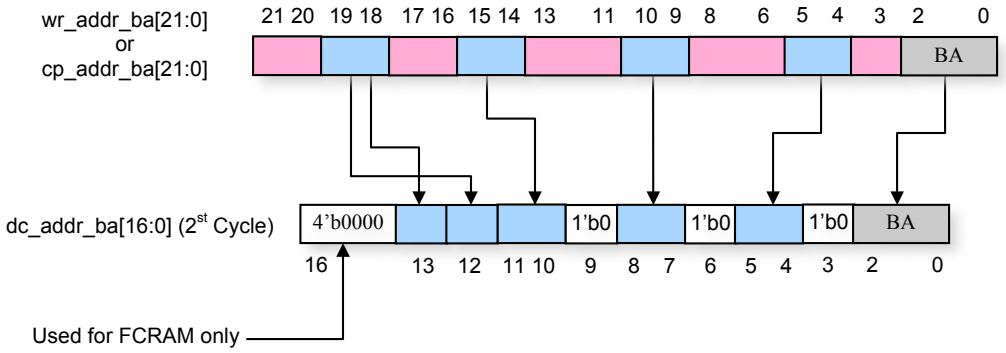
When data is written to the memory through a CPU access, it comes from the cp_addr_ba[21:0] register, which is already a 22-bit cell address:



The RLDRAM device operates in multiplexed address mode in order to save on pin count. This means the address is sent in two cycles, requiring only 11 physical address pins to be used for the 19-bit address (an additional 3 pins are still required for the bank address).

The DMC takes care of the address muxing and splits the address into two cycles before it passes it to the Phy:

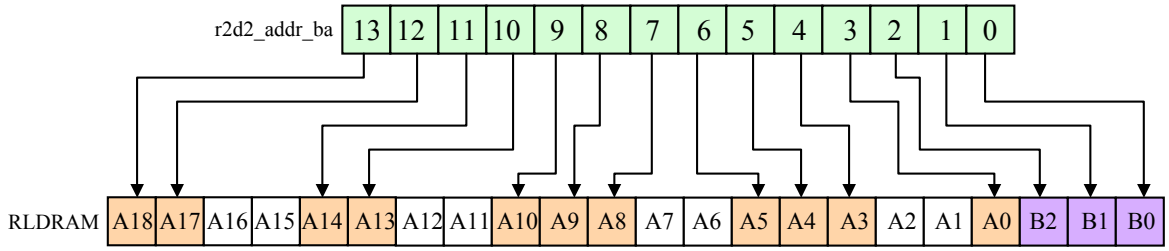




The address and bank addresses are SDR signals, so the Phy passes these buses out to the pins of R2D2 without modifying the format.

The physical connections to the memory pins are as follows:

addr_ba[2:0] = bank address → Micron {B2, B1, B0}
addr_ba[13:3] = address → Micron {A18, A17, A14, A13, A10, A9, A8, A5, A4, A3, A0}



During the MRS commands, we program 22'h1550 into cp_addr_ba. This will translate to 17'h150 on the Ax cycle, and 17'h110 on the Ay cycle.

4 R2D2 KEY REGISTER SUMMARY

The registers most often used in diagnostics and devtest are listed here with their descriptions:

Register Name	Address	Bits	Function	Default Value
DH_CPU_REG_MOD_RESET__2	11'h006	16	Per-module soft resets [15] – PB2N [14] – PB2AR [13] – OCT [12] – 10G MAC loopback [11] – N2PB [10] – MSTAT [9] – MACIF [8] – LM [7] – LANIF [6] – GXSYNC [5] – RTP_FDR [4] – CPU [3] – CBL	16'hEFFF (All modules in soft reset; 10G MAC loopback disabled)

Register Name	Address	Bits	Function	Default Value
			[2] – AROC [1] – ARIC [0] – AR2PB	
DH_CPU_REG_MOD_RESET__1	11'h007	16	[15] – DMC_RX [14] – TXF [13] – CDL [12] – MIIM [11] – PLL [10] – FM [9] – TTRUNK [8] – CMN_CFG [7] – TFIFO [6] – TMC [5] – TLB [4] – RTP [3] – RMC [2] – PQ [1] – PPC [0] – PORTIF	16'hF7FF (All modules except the PLL in soft reset)
DH_CPU_REG_MOD_RESET__0	11'h008	16	[15:2] – *Reserved* [1] – ELAM [0] – DMC_TX	16'h0003 (Both modules in soft reset)
DH_PQ_REG_DESCRPTR_MEM_MAP	11'h03F	5	[4:0] – Memory map for the packet descriptors in the external memory. The value of this register determines which 16MB space in external memory holds the descriptors. The legal values of this register are 0-31 for Dhruva (512MB) and 0-7 for R2D2 (128MB). Mapped to the lowest 16MB of memory by default.	5'h00 (Descriptors in external memory mapped to lowest 16MB space)
DH_N2PB_CP_RX_CFG_NP_PA	11'h075	16	[1] – Multicast queue disable: default is for the multicast queue to be enabled for each port; all multicast queues are disabled by setting this bit. [0] – Packet packing disable: "0" is the default value which enables the packing of last 32 bytes of the current packet with the first 32 bytes of the next packet of the queue, in a single cell. "1" disables this function, forcing new packets to always start at a cell boundary.	16'h0000 (Multicast queues and packing enabled)
DH_N2PB_CP_PBR_DIV_INDX DH_N2PB_CP_PBR_DIV	11'h076 11'h077	16	Indicates start and ending 512B address (8 cells) of each queue in Rx memory. 64B cell address is determined by left-shifting this number by 3. Indices 0-38 indicate the starting address of that particular queue. Index 39 is the ending address of queue 38.	Default starting address for queue 0 is 20'h08000 (i.e. first 16MB are reserved for descriptors). Each queue is reserved 20'h01000 (2MB). Thus, the ending address of queue 38 is 20'h2F000 (94MB).
DH_PB2AR_CP_RX_CFG_PA	11'h109	16	[0] – Disables read from PB2AR FIFO (Rx memory): only used for debug purposes.	16'h0000 (Read from PB2AR FIFO enabled)
DH_AR2PB_CP_TX_CFG_AP	11'h135	16	[8] – R2D2 mode "0" = Dhruva mode, 27 open pages (1 for every 2 queues + multicast) "1" = R2D2 mode: 53 open pages (1 per Tx queue+ multicast) [6] – When in Dhruva mode, this bit distinguishes between single open-page Tx mode and 27 open-page mode. "0" = 27 open-page mode (default) "1" = single open-page mode	9'h100 (R2D2 mode)

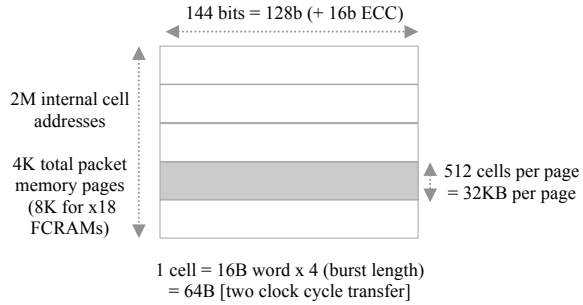
Register Name	Address	Bits	Function	Default Value
DH_AR2PB_CP_TX_CFG_AP_PN	11'h136	2	[1] – Select for port 0 thru port 12. "0" = groups q0, q1 & q2 in one page, q3 in another page "1" = groups q0/q1 in one page & q2/q3 in another. [0] – Compact header mode "0" = normal header mode – DBUS header is written to external memory along with the packet "1" = compact header mode –Depending on some criteria (non-ISL, non-Inband packets etc.), the DBUS header of the incoming packet may not be written into the memory in order to optimize the memory bandwidth utilization for certain packet sizes. All necessary information from the DBUS header is extracted and stored in the packet descriptor and then used to recreate the DBUS header in a downstream module. (default)	2'b01 (Compact header mode)
DH_AR2PB_CP_AP_RED_ENB	11'h188	14	[13] – Enable drops due to maximum page allowances (default)	(Page quota drops enabled)
DH_PB2N_CP_TX_CFG_PN	11'h198	12	[0] – Halt reads from Tx memory by disabling the service of PQ descriptors for all queues.	12'h0 (Reads from Tx memory enabled)
DH_ARIC_DH_ARIC_ADDR → DH_ARIC_SPAN_DVLNF_CAPX_ACPT_00 (6'h00) → DH_ARIC_SPAN_DVLNF_CAPX_ACPT_01 (6'h02) → DH_ARIC_SPAN_DVLNF_CAPX_ACPT_02 (6'h04) → DH_ARIC_SPAN_DVLNF_CAPX_ACPT_03 (6'h06) → DH_ARIC_SPAN_DVLNF_CAPX_ACPT_04 (6'h08) → DH_ARIC_SPAN_DVLNF_CAPX_ACPT_05 (6'h0A) → DH_ARIC_SPAN_DVLNF_CAPX_ACPT_06 (6'h0C) → DH_ARIC_SPAN_DVLNF_CAPX_ACPT_07 (6'h0E) → DH_ARIC_SPAN_DVLNF_CAPX_ACPT_08 (6'h10) → DH_ARIC_SPAN_DVLNF_CAPX_ACPT_09 (6'h12) → DH_ARIC_SPAN_DVLNF_CAPX_ACPT_10 (6'h14) → DH_ARIC_SPAN_DVLNF_CAPX_ACPT_11 (6'h16) → DH_ARIC_SPAN_DVLNF_CAPX_ACPT_12 (6'h18) DH_ARIC_DH_ARIC_DATA	11'h1E3 11'h1E4	11	[0] – Force ARIC to accept all packets for this particular port. "0" = Disable packet accept (default) "1" = Enable packet accept	11'h0 (ARIC packet accept disabled)
DH_ARIC_DH_ARIC_ADDR → DH_ARIC_CP_CRC_CHK_PAD_EN (6'h27) DH_ARIC_DH_ARIC_DATA	11'h1E3 11'h1E4	11	[2] – Enable Ethernet packet data CRC check (default)	5'h07 (Enable Ethernet packet data CRC check)
DH_ARIC_CP_ARGS_FLOW_CTRL	11'h261	3	Indication of flow control from Metropolis/Hyperion (active low): [2] – High priority flow control [1] – Low priority flow control [0] – Multicast traffic flow control	Status Register
DH_ARIC_CP_ARIC_INTERNAL_LPBK	11'h264	1	Enable Metropolis interface loopback.	1'b0 (Metropolis loopback disabled)
DH_CMN_CFG_DH_CMFG_ADDR → DH_CMN_CFG_TOS2Q_63_OPMD (8'h0F) DH_CMN_CFG_DH_CMFG_DATA	11'h26D 11'h26E	16	[10] – 10G mode: "0" = 10/100/1000 Mbps "1" = 10 Gbps	16'h0 (10/100/1000 Mbps mode) Write 16'h0400 for 10G mode

Register Name	Address	Bits	Function	Default Value
DH_CMN_CFG_DH_CMFG_ADDR → DH_CMN_CFG_MISC_HYPERION (8'h0A) DH_CMN_CFG_DH_CMFG_DATA	11'h26D 11'h26E	16	[6] - force/expect packets to/of be of 32-byte boundary [5:3] – AROC FIFO full threshold: when this number of lines are filled, AROC FIFO full is asserted. [2] – Enables inband mode for port 12. This is the default mode. [1] – Enables inband mode for port 11. Default is normal mode. [0] – Enables Hyperion mode; "0" is the default value, which enables Argos/Metropolis mode.	7'h7C (Argos/Metropolis mode enabled, port 12 in inband mode, port 11 in normal mode) Write 7'h7D to enable Hyperion mode.
DH_CMN_CFG_DH_CMFG_ADDR → DH_CMN_CFG_CP_TRUNK_MD_00(8'h44) → DH_CMN_CFG_CP_TRUNK_MD_01(8'h45) → DH_CMN_CFG_CP_TRUNK_MD_02(8'h46) → DH_CMN_CFG_CP_TRUNK_MD_03(8'h47) → DH_CMN_CFG_CP_TRUNK_MD_04(8'h48) → DH_CMN_CFG_CP_TRUNK_MD_05(8'h49) → DH_CMN_CFG_CP_TRUNK_MD_06(8'h4A) → DH_CMN_CFG_CP_TRUNK_MD_07(8'h4B) → DH_CMN_CFG_CP_TRUNK_MD_08(8'h4C) → DH_CMN_CFG_CP_TRUNK_MD_09(8'h4D) → DH_CMN_CFG_CP_TRUNK_MD_10(8'h4E) → DH_CMN_CFG_CP_TRUNK_MD_11(8'h4F) → DH_CMN_CFG_CP_TRUNK_MD_12(8'h50) DH_CMN_CFG_DH_CMFG_DATA	11'h26D 11'h26E	6	[5:0] – Controls the trunking mode of each port: Bit position: 0: 802.1p 1: native 2: trunk mode 3: trunk enable 4: non-tag 5: tag Rx trunking uses bits [5:1] Tx trunking uses bits [3:0] 00_0xxx = [DH_RX_NRML_MD] (Rx & Tx) 00_100x = [DH_RX_ISL_MD] (Rx & Tx) 00_101x = [DH_RX_ISLN_MD] (Rx & Tx) 00_1100 = [DH_RX_1Q_MD] (Rx & Tx) 00_1101 = [DH_RX_1P_MD] (Tx only) 00_1110 = [DH_RX_1QN_MD] (Rx & Tx) 00_1111 = [DH_RX_1PN_MD] (Tx only) 01_xxxx = [DH_RX_TAGN_MD] (Rx only) 10_xxxx = [DH_RX_TAG_MD] (Rx only) 11_xxxx = [DH_RX_AUTO_MD] (Rx only)	6'h30 (Auto mode for Rx, Normal mode for Tx)
DH_CMN_CFG_DDR_CHK_RD_WR_CTRL	11'h2FF	4	[2] – Disable read from the AROC sync-FIFO to allow packets to be built there. [0] – Disable read from ARIC sync-FIFO: This bit is set to hold data in the ARIC sync-FIFO to allow for a CPU read (16 cycles required to read a complete line).	4'h0 (Reads from ARIC and AROC FIFOs enabled)
DH_LANIF_DH_LANIF_INDX → DH_LANIF_DH_MAC_RESET_XX (9'h02X), where X is the port number DH_LANIF_DH_LANIF_DATA	11'h32F 11'h330	16	Resets for the internal MAC cores: [4] – Rx MAC Reset (1000Mbps) [3] – Tx MAC Reset (1000Mbps) [2] – Rx MAC Reset (10/100Mbps) [1] – Tx MAC Reset (10/100Mbps) [0] – Link Reset	5'h06 (10/100 MACs in reset)
DH_LANIF_DH_LANIF_INDX → DH_LANIF_DH_GMAC_CFG_XX (9'h03X), where X is the port number DH_LANIF_DH_LANIF_DATA	11'h32F 11'h330	16	[6:5] – MAC mode "00" = 1000 Mbps (default) "01" = 10 Mbps "10" = 100 Mbps "11" = 1000 Mbps [4] – GMII mode "0" = 8b/10b encoded (TBI) mode "1" = GMII mode (default) [3] – No SERDES mode "0" = SERDES mode (default) "1" = No SERDES mode – used when in TBI mode and not connecting to a SERDES device, i.e. for MAC to MAC connections [2] – Hardware auto-negotiation "0" = disabled (default) "1" = enabled [1] – Force Rx Link [0] – Force Tx Link	16'h0010 (1Gbps, GMII mode, SERDES mode, no hardware auto-negotiation, neither link forced) Note: GMII mode with No SERDES ([4:3] = "11") is not valid.

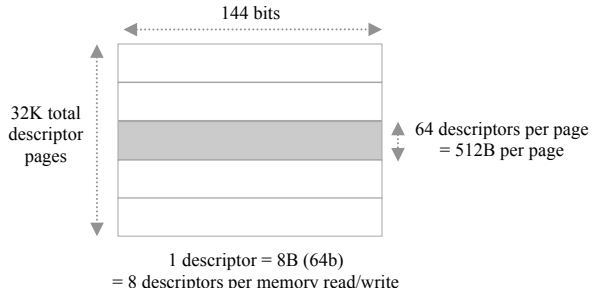
Register Name	Address	Bits	Function	Default Value
DH_TMC_CP_TMC_CFG	11'h47E	14	[13] – Store packet descriptors in Tx memory	14'h1C53 (Descriptors stored in Rx memory)

5 R2D2 MEMORY MANAGEMENT QUICK REFERENCE

External Packet Buffer (4 x36 FCRAMs)
128MB Total Memory (144MB Physical)

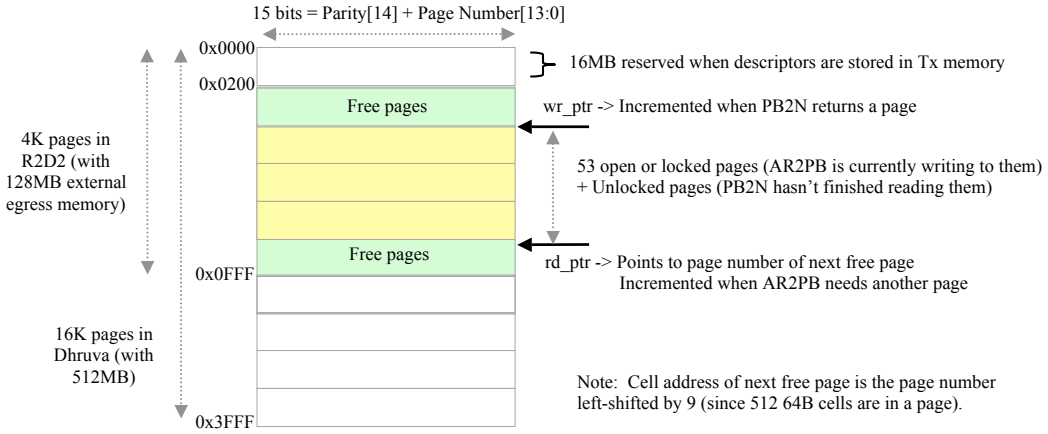


External Descriptor Storage
16MB of external packet buffer reserved for descriptors (holds 2M descriptors).



Note: Cell address of next free descriptor page is the page number left-shifted by 3 (since 8 64B cells are in a page).

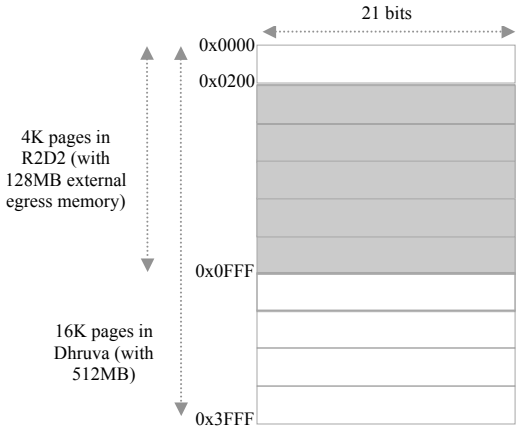
FM Internal SRAM
Circular-FIFO based memory to maintain list of free packet data pages



Note: Cell address of next free page is the page number left-shifted by 9 (since 512 64B cells are in a page).

PPC Internal SRAM

Maintains counts of outstanding cells on per-page basis



512 pages reserved when descriptors are stored in Tx memory

AR2PB maintains a cell count for each open page while it's writing data to it. When the entire page is written, it instructs PPC to increment its count by this amount.

PB2N reads out packets and instructs PPC to decrement the count by the number of cells in the packet just read out.

Thus, a negative value in PPC indicates PB2N has already read data out of a page, but the page is not full yet, so AR2PB hasn't unlocked it. This is a normal condition.

A positive value in PPC indicates there are packets remaining in the queues.

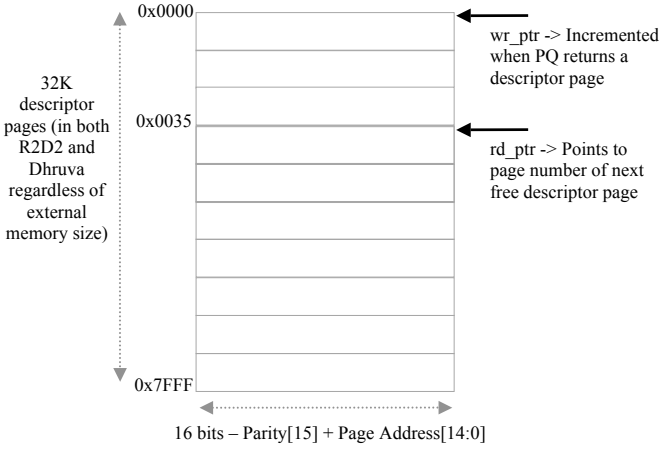
Parity[20], Page ID[19:14], Valid[13], Cell Count[12:0]

Note: Diags reads the PPC memory into 2 16-bit registers. Bits[20:5] are placed into the MSB register. Bits[4:0] are 0-padded and placed into the LSB register.

LM Internal SRAMs

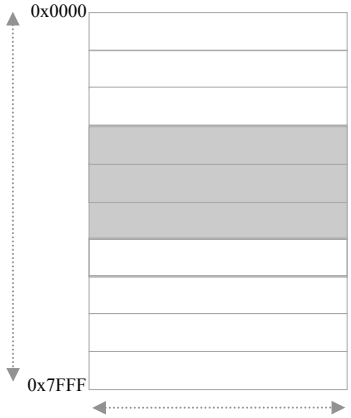
Maintains linked list and free list of packet descriptors

Free List SRAM



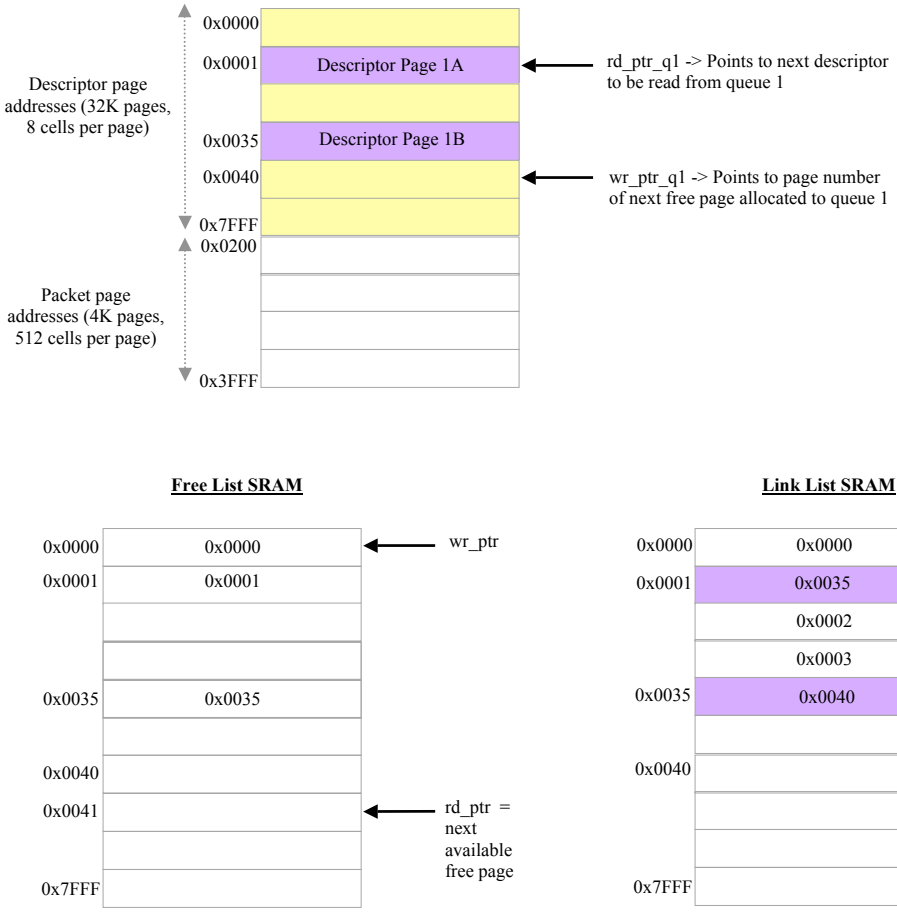
16 bits – Parity[15] + Page Address[14:0]

Link List SRAM



16 bits – Parity[15] + Page Address[14:0]

6 EXTERNAL DESCRIPTOR MANAGEMENT EXAMPLE



7 R2D2 DIAGNOSTICS BASIC COMMAND LIST

Check supervisor card images:	rommon> dir disk0:
Check line card images:	rommon> dir bootflash:
Load supervisor card image:	rommon> boot disk0:supw_lepus8p
Load only image skipping POD:	rommon> boot disk0: -p
Show global parameters:	show
Show test specific parameters:	show –tr2d2_1
Run register tests:	run –tr2d2_1
View register test result:	result –tr2d2_1
Run Tx & Rx GMTL tests:	run –tr2d2_6 –tr2d2_7
Only run “address” GMTL test:	set –tr2d2_6 –pmode=address
List all Tx DMC counters:	ra –dr2d2 –o548 –l7
List all Rx DMC counters:	ra –dr2d2 –o5b7 –l7
Write to Tx external memory:	ma –dr2d2 –mtxpb –b<addr> -as –v<wdata>
Write to Rx external memory:	ma –dr2d2 –mrxpb –b<addr> -as –v<wdata>
Read from Tx external memory:	ma –dr2d2 –mtxpb –b<addr> -ad
Read from Rx external memory:	ma –dr2d2 –mrxpb –b<addr> -ad
Perform register read/modify/write:	ra –dr2d2 –o<addr> -am
View all counters on port 1:	dump –dr2d2 counters –p1
View all errors on one screen:	dump –dr2d2 errors –y
List all registers with regex:	ra –dr2d2 –al –vFREELIST
Read indirect register:	ra –dr2d2 –b1249 -ad

8 SANITY CHECKS TO BE PERFORMED AFTER EACH TEST

	Name	Description	Command	Comments
Tx Path	PQC	Ensure that cellcount for each queue returns to initial value	**ma -dr2d2 -mpqc -b0 -l34**	
	LMPtr	Ensure that the head and tail pointers for each queue are equal	**ma -dr2d2 -mpdqh -b0 -l34** **ma -dr2d2 -mpdqt -b0 -l34**	
	LLC	Ensure that the count of pointers in each queue is zero	**ma -dr2d2 -mllc -b0 -l34**	
	LM_FREELIST_CNT	Ensure that the total number of pointers returns to the initial	**TCL proc LmFree**	
	FM_FREELIST_CNT	Ensure that the total number of free pages returns to the initial value	**TCL proc FmFree**	
Rx Path	N2PB_FREE_CNT	Ensure that the n2pb free counts for all queues return to their initial values	**ra -dr2d2 -b165 -l39**	The -b165 is the starting address of the first cp_np_free_cnt_## register. This value may change depending on the version of the image you're using. Use **ra -dr2d2 -al -vcp_np_free_cnt** to find the correct value to use.
General	Counters	Account for all counter values	**dump -dr2d2 counters -y**	
	Drops	Account for all drop counter values	**dump -dr2d2 drop -y**	
	Errors	Check error counters	**dump -dr2d2 errors -y**	
	Interrupts	Check interrupts	**dump -dr2d2 intr**	

9 LAB DEBUG CHEAT SHEET

9.1 Ingress Drop Counters

First, determine where the drop occurred by observing all relevant per-port and global drop counters. Note that all of these drops occur prior to data writes to the external memories.

DH_RTP_MORE_SZ_XX: # of packets greater than 1518B received by RTP (after header de-encapsulation)
DH_RTP_LESS_SZ_XX: # of packets less than 64B received by RTP (after header de-encapsulation)
DH_N2PB_CP_NP_DROP_TBL_IDX_REG
DH_N2PB_CP_NP_DRP_TBL_DT0_RG – upper 16 bits of WRED drop counter
DH_N2PB_CP_NP_DRP_TBL_DT1_RG – lower 16 bits of WRED drop counter

RDROP: number of packets dropped in the receive path (see below)

The RDROP counter may be incremented for various reasons. The following 40-bit vector is sent from several different modules to the MSTAT module. Each bit corresponds to a counter that gets incremented. As listed below, however, several bits will cause the RDROP counter to be incremented – rt_stat_val[39:0]:

[39]-bad_encap
[38]-islfrm - ISL_1P_1Q_INB
[37]-q1frm - ISL_1P_1Q_INB
[36]-p1frm - ISL_1P_1Q_INB
[35]-inbfrm - ISL_1P_1Q_INB
[34]-wanfrm - RCODE
[33]-bcfrm - BCAST
[32]-mcfrm - MCAST
[31]-ucfrm - UCAST
[30]-drop_bs - RDROP
[29]-drop_ms - RDROP
[28]-drop_us - RDROP
[27:25]-cosval[2:0]
[24:9]-mod_frmlength[15:0]
[8]-ctrlfrm – RCTL
[7]-pausefrm - RPAUSE
[6]-crcerr – RCRC – (outer CRC error from MAC) or (ISL frame & inner CRC error)
[5]-symerr - RCODE
[4]-seqerr - RCODE
[3]-cbldrop - RDROP
[2]-crcdrop- RDROP – (outer CRC error from MAC) & cp_crc_chk_en_xx & not a WAN frame
[1]-isldrop – RDROP – ISL frame &
((inner CRC error & cp_isl_crc_chk_en) |
(dbus[143:128] !=AAAA03 & cp_isl_con_chk_en) |
(length in ISL header != actual frame length & cp_isl_len_chk_en))
[0]-inbdrop - RDROP

All mentioned configuration registers are indirectly accessed through RTP_DATA & RTP_ADDR:

cp_crc_chk_en_00 = DH_RTP_CRCCHK_ISLCHK[0]

cp_isl_crc_chk_en_00 = DH_RTP_IPO_NETB_TR_RSMP_FPOEUPR_00[14]
cp_isl_con_chk_en = DH_RTP_CRCCHK_ISLCHK[15]
cp_isl_len_chk_en = DH_RTP_CRCCHK_ISLCHK[14]

9.2 Ingress Data Path Counters

All relevant data path counters should be dumped to isolate the problem to a particular area of the chip. The appropriate counters are listed below in order of data flow through the ingress path.

LANIF Counters:
DH_MSTAT_CP_STAT_ADDR_CTRL:
RTOT: total number of frames received by the MAC, including bad frames
RTOT_UCAST: total number of unicast frames received by the MAC
RTOT_MCAST: total number of multicast frames received by the MAC
RTOT_BCAST: total number of broadcast frames received by the MAC
R64: updated when frame length is 64 bytes (including bad-CRC frames)
R127: updated when frame length is between 65-127B (including bad CRC frames)
R255: updated when frame length is between 128-255B (including bad CRC frames)
R511: updated when frame length is between 256-511B (including bad CRC frames)
R1032: updated when frame length is between 512-1032B (including bad CRC frames)
R1518: updated when frame length is between 1033-1518B (including bad CRC frames)
R1548: updated when frame length is between 1519-1548 (including bad CRC frames)
RGIANT: updated when frame length is > 1548 and CRC is good
SHORT: updated when frame length is < 64B and CRC is good
RUNT: updated when frame length is < 64B and CRC is bad
JABBER: updated when frame length is > 1518B and CRC is bad
RCRC: received frames with bad CRC that are not runt or jabber – may or may not get dropped depending on register configurations

RTP Counters:
DH_RTP_RTP_IN_PKT_CNT: # of packets from RTP_FDR to RTP

N2PB Counters:
DH_N2PB_N2PB_IN_PKT_CNT: # of packets from RTP to N2PB

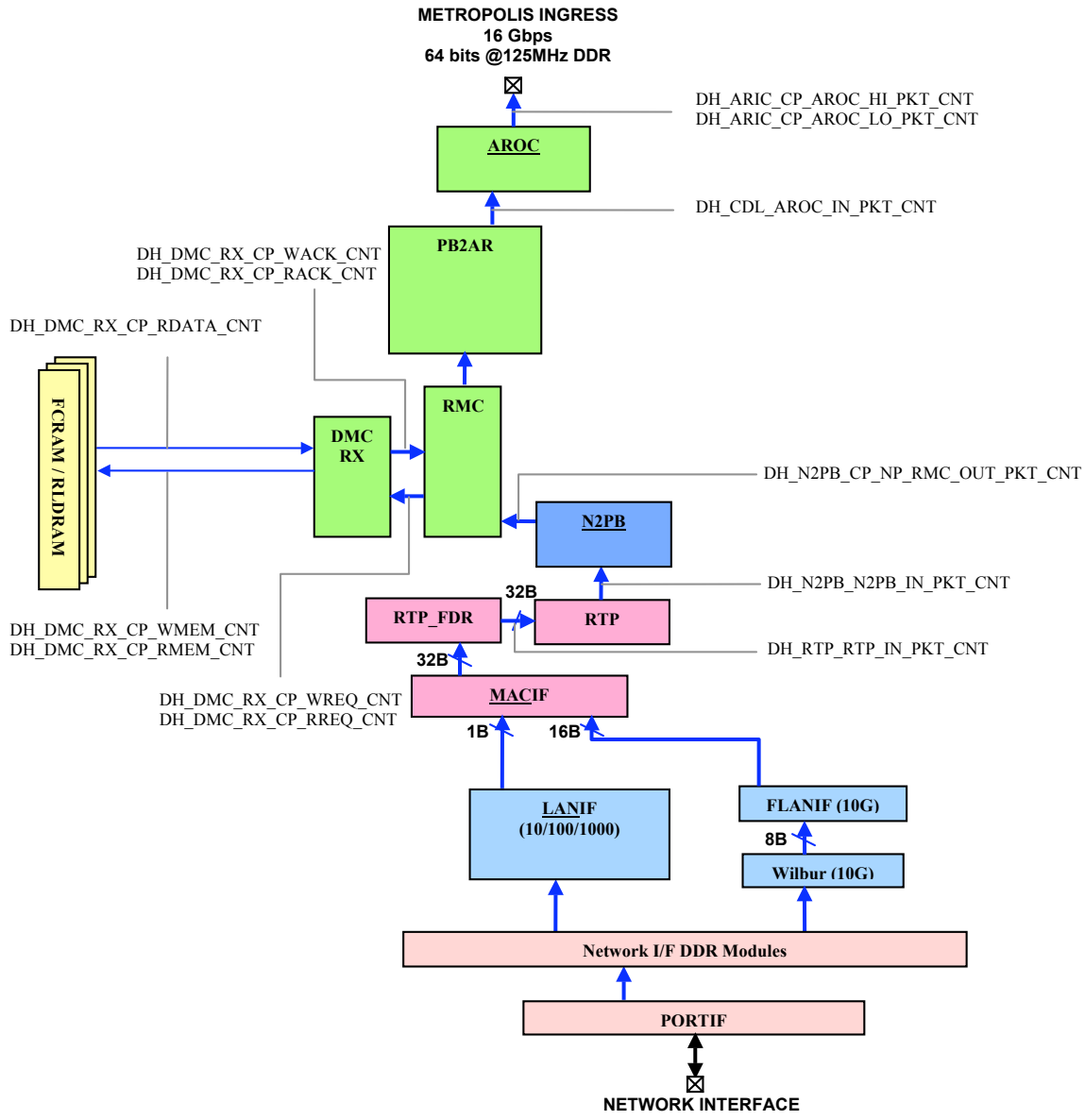
RMC Counters:
DH_N2PB_CP_NP_RMC_OUT_PKT_CNT: # of packets from N2PB to RMC

DMC Counters:
DH_DMC_RX_CP_WREQ_CNT: # of write request cells from RMC to the DMC
DH_DMC_RX_CP_WMEM_CNT: # of write request cells from DMC to external memory
DH_DMC_RX_CP_WACK_CNT: # of write acknowledgement cells from DMC to RMC
DH_DMC_RX_CP_RREQ_CNT: # of read request cells from RMC to the DMC
DH_DMC_RX_CP_RMEM_CNT: # of read request cells from the DMC to external memory
DH_DMC_RX_CP_RDATA_CNT: # of read data cells returned from external memory
DH_DMC_RX_CP_RACK_CNT: # of read acknowledgement cells from the DMC to RMC

AROC Counters:
DH_CDL_AROC_IN_PKT_CNT: # of packets from PB2AR to AROC

ASIC Interface Counters:
DH_ARIC_CP_AROC_HI_PKT_CNT: # high priority packets going from AROC to MET/HYP

DH_ARIC_CP_AROC_LO_PKT_CNT: # low priority packets going from AROC to MET/HYP



9.3 Ingress Miscellaneous Counters

The following counters could indicate a 10G problem with incorrectly programmed phy, cable issues, or xenpak issues:

DH_GXSYNC_GS_CODE_ERR
DH_GSXYNC_GS_CODE_VIO
DH_GXSYNC_GS_RIPG_VIO

9.4 Egress Drop Counters

R2D2_AR2PB_CP_TXQ0_MINPG_DROP_CNT: # drops due to max pages per queue exceeded (based on DH_PQ_CP_PQ_PMAXPG register value)

DH_PQ_CP_REG_DROPCNT: # drops due to max descriptor pages per queue exceeded (based on DH_PQ_CP_MAXPG register value). In diags this is called:

R2D2_PKT_DESC_Q_LNK_LIST_DROP_CNTR:

WRED and DT drops are based on PQ_CP_PQ_CELLCNT:

PMAXPG: 4K – 512 memory pages available. Since 53 can be open at any one time (1 per queue plus 1 for multicast traffic), the fair share is (4096 – 512) / 53 = 67 pages per agent. We'll likely want to set this value to some amount greater than the fair share.

For Lepus, we only use 12 ports, giving us 48 queues and 49 agents. Fair share would be 73 pages per agent.

PMINPG: How many pages do you want to guarantee to each queue? Dhruva default is 16.

MAXPG: R2D2 supports 32k descriptor pages shared by 52 queues, so the fair share is 630 descriptor pages per queue.

For Lepus, due to a bug in the memory wrapper (aka minpg bug), we only use 16k descriptor pages, so fair share is 315 descriptor pages.

MINPG: How many descriptor pages do you want to guarantee to each queue? Dhruva default is 8.

CELLCNT: How many cells can each queue hold before issuing drops? Ideally, for unicast traffic, we expect to perform WRED or DT drops before maxpg drops. The fair share for each queue is 128MB – 16MB (descriptor storage) / 52 = 2.2MB. Since each cell is 64B, this corresponds to about 35K cells.

9.5 Egress Data Path Counters

All relevant data path counters should be dumped to isolate the problem to a particular area of the chip. The appropriate counters are listed below in order of data flow through the egress path.

R2D2_ARIC_ARIC_IN_PKT_CNT: # of packets received by ARIC from MET/HYP (32-bit)
R2D2_ARIC_CP_ARIC_PKT_CNT: duplicate 32-bit register to the one above

R2D2_AR2PB_CP_AP_ARIC_IN_PKT_CNTR: # of packets entering AR2PB from ARIC
R2D2_AR2PB_CP_AP_TMC_OUT _PKT_CNTR: # of packets leaving AR2PB to TMC

R2D2_DMC_TX_CP_WREQ_CNT: # of write request cells from RMC to the DMC
R2D2_DMC_TX_CP_WMEM_CNT: # of write request cells from DMC to external memory
R2D2_DMC_TX_CP_WACK_CNT: # of write acknowledgement cells from DMC to RMC

R2D2_PB2N_CP_PN_TMC_OUT_PKT_CNTR: # of packet requests from PB2N to TMC

R2D2_DMC_TX_CP_RREQ_CNT: # of read request cells from TMC to the DMC
R2D2_DMC_TX_CP_RMEM_CNT: # of read request cells from the DMC to FCRAMs
R2D2_DMC_TX_CP_RDATA_CNT: # of read data cells returned from FCRAMs
R2D2_DMC_TX_CP_RACK_CNT: # of read acknowledgement cells from the DMC to TMC

R2D2_PB2N_CP_PN_TTRUNK_OUT_PKT_CNTR: # of packets leaving PB2N to TTRUNK
R2D2_TLB_TTRUNK_IN_PKT_CNT: # of packets entering TTRUNK from PB2N

10 DMC KILLER PACKET DEFINITION

The following 128B packet was defined to maximize the number of bit changes from posedge to posedge and negedge to negedge transitions.

IPO is set, length is 0x60, FPOE is 0xFF, L3 is 1E00, DBUS FCS is 697C, DBUS CRC is 41B8 F9EB:

```
0002 0000 0000 0000 0000 0000 6000 0000 1E00 0000 0000 00FF 0000 0000 0000 697C
0000 0000 FFFD FFFF FFFF FFFF FFFF FFFF 9FFF FFFF E1FF FFFF FFFF FF00 FFFF FFFF

(ECC1 is then 8611 06C8 070C 1106)
(~ECC1 is then 79EE F937 F8F3 EEF9)

0002 0000 0000 0000 0000 0000 6000 0000 1E00 0000 0000 00FF 0000 0000 79EE F937
F8F3 EEF9 FFFD FFFF FFFF FFFF FFFF FFFF 9FFF FFFF E1FF FFFF FFFF FF00 41B8 F9EB

(ECC2 is then 8611 0633 710C 1114)
(~ECC2 is then 79EE F9CC 8EF3 EEEB)

CRC is calculated using the Ixia, while the FCS and ECC are calculated using
Verilog code:  /auto/hss0/c2cr/asic/users/ktomei/r2d2_current/ecc_calc/

Rearranged to show the cycle transitions, the packet looks like this when
written to the FCRAMs:

0002 0000 0000 0000 0000 0000 6000 0000 1E00 0000 0000 00FF 0000 0000 0000 697C 0000 0000
FFFD FFFF FFFF FFFF FFFF FFFF 9FFF FFFF E1FF FFFF FFFF FF00 FFFF FFFF (8611 06C8 070C 1106)
0002 0000 0000 0000 0000 0000 6000 0000 1E00 0000 0000 00FF 0000 0000 79EE F937 F8F3 EEF9
FFFD FFFF FFFF FFFF FFFF FFFF 9FFF FFFF E1FF FFFF FFFF FF00 41B8 F9EB (8611 0633 710C 1114)

The coverage is as follows:

1st cycle transition (1 to 2):  288 bits total, 1st 224 bits are flipped
2nd cycle transition (2 to 3):  288 bits total, ALL bits are flipped
3rd cycle transition (3 to 4):  288 bits total, 1st 196 bits are flipped
4th cycle transition (4 to 1):  288 bits total, 1st 196 bits are flipped

Thus, the bits in Phys 3-7 have complete coverage.   The bits in Phys 0-2 have
slightly less coverage, but are guaranteed to transition at least once for each
packet.

The following configurations should be made to preserve the packet:

$MetLoopback enable
$r2d2_reg_wr 0x31E 0x405 (place ports 0 & 2 in inband mode)
$r2d2_reg_wr 0x7C5 0x5 (place ports 0 & 2 in symmetric CRC mode)
$r2d2_reg_wr $R2D2_ARIC_DH_ARIC_ADDR 0x27 (CRC check in ARIC)
$r2d2_reg_wr $R2D2_ARIC_DH_ARIC_DATA 0x5 (enable the CRC check)
```

Field	Description	Value	Explanation	New Value	Explanation
255	1 MPLS label	1'b0		1'b0	
254	>1 MPLS label	1'b0		1'b0	
253	IP/IPX valid	1'b0		1'b0	
252	IP checksum valid	1'b0		1'b0	
251	L4 header valid	1'b0		1'b0	
250:248	COS	3'b101	?	3'b0	High priority packet
247	COS type	1'b1	?	1'b0	

Field	Description	Value	Explanation	New Value	Explanation
246:243	Frame type	4'b1111	?	4'b0	
242	802.1p type	1'b1	?	1'b0	
241	IPO	1'b0	?	1'b1	IP Field Override
240	No establish	1'b0		1'b0	
239	Control[15] - No statistics	1'b0		1'b0	
	Control[14:11] = {	4'b0		4'b0	
	Do not use,				
238:235	Do not use,				
	Do not use,				
	Central rewrite}				
234	Control[10] == Trusted	1'b0		1'b0	
233	Control[9] == Notify index learn	1'b0		1'b0	
	Control[8:5] = {	4'b0		4'b0	
	Notify New Learn,				
232:229	Disable New Learn,				
	Disable Index Learn,				
	Don't Forward				
	}				
228	Control[4] == Index Direct	1'b0		1'b0	
	Control[3:1] == {	3'b0		3'b0	
	Don't Learn,				
227:225	Conditional Learn,				
	Bundle Bypass				
	}				
224	Control[0] == TIC, Qos Trusted input class	1'b0		1'b0	
223:221	*Reserved*	3'b0		3'b0	
220	Truncated	1'b0		1'b0	
219:208	VLAN	12'hDEA		12'h000	
207	Netflow Inband			1'b0	
206	Ignore QOS0			1'b0	
205	Ignore QOS1			1'b0	
204	Apply ACL0			1'b0	
203	Ignore ACL			1'b0	
199:196	*Reserved*			4'b0	
195	Source flood			1'b0	
194:176	Source index			19'h00000	
194:191	Source Index[18:15] == Card Instance				
190:176	Source Index[14:0] == port index				
175:172	Rx SPAN			4'b0	
175:174	RX SPAN[3:2]				
173:172	RX SPAN[1:0]				
171	*Reserved*			1'b0	
170:168	Bundle Port			3'b0	
167:152	Length			16'h0060	
151:144	Status				
151	Bad CRC			1'b0	
150	Length < 64B			1'b0	
149	Length > 1518B			1'b0	
148	ISL frame			1'b0	
147	802.1q frame			1'b0	
146	*Reserved*			1'b0	
145	BPDU class frame			1'b0	
144	*Reserved*			1'b0	
143:128	ISL Reserved			16'h0	
127:112	Layer 3			16'h0000	
127:125	Encoded frame type				
124:121	Encoded protocol type				
120	*Reserved*				
119:112	Protocol type				

Field	Description	Value	Explanation	New Value	Explanation
111:40					
111:96	Result				
103	QT bit				
95:92	L3 learning				
91:88	Card type				
87:85	*Reserved*				
84	RMON sampling				
83:64	FPOE				
63:40	RMON tag				
39:36	Protocol mask				
35	Destination flood				
34:16	Destination index				
34:31	Destination index[18:15]				
30:28	Destination index[14:12]				
27:16	Destination index[11:0]				
15:0	FCS				

11 GATE AND INSTANCE COUNTS

This section includes a per-subchip pre-layout gate and instance count. Technology was TSMC .13um.

Sub-chip	Pre-layout Gate Count	Pre-layout Instance Count
ARIC	**637K**	**222K**
AROC	**862K**	**266K**
LAN	**491K**	**177K**
MAC	**960K**	**325K**
N2PB	**605K**	**241K**
PB2N	**440K**	**138K**
PQ	**747K**	**242K**
TMC	**609K**	**175K**

Note, the DMC contains approximately 90k bits of RA.

R2D2 Egress Queue Mapping

SOP Cycle:

If ((Ports 11 or 12 are selected) || (1G Mode) || (q_sel_tx_rx != 4'b10xx))
    Queue = {tos_cos_vld, tos_cos_val}
Else // 10G mode && Ports 11 and 12 are NOT selected && VLAN2Q mode
    Queue = {2'b0, queue_no_int}

1G port select (cp_op_mode == 2'b00): psel_xx (final port select after filtering)
10G port select (cp_op_mode == 2'b01): {psel_xx[12:11], original incoming FPOE for ports [10:1] psel_xx[0]}

Cp_op_mode = {1'b0, tos2q_63_opmd_fp[10]}
    -> default value is from strap pin, but if cp_tx_cfg_ap[7] is set (off by default), then the register value is used (tos2q_63_opmd[10])