D A T A B O O K

Version 1.01
July 2003

INIC-2430

FireWire 800 to ATA Bridge IC

initio

Revision Notes
for INIC-2430 Data Book: P/N 2430X-011DS Rev 1.01, 7/03

Revision History

Doc Rev 1.00	(P/N X2430X-011DS)	06/03
Doc Rev 1.01	(P/N 2430X-011DS)	07/03 (current release)

Technical Information Changes in this Data Book Release

The table below lists technical information that has changed from the *INIC-2430 Data Book* revision **1.00** to the *INIC-2430 Data Book* revision **1.01**. (Superficial or non-technical edits are not indicated.)

Section Updated	Pages Affected	Change Description
5.4	64	Table 5-4, ICC values *were* tbd.
6.1	65	TLSU, TLH, and TLHZ timings changed to 1.5 minimum, and 2.9 maximum. For TLSU, TLH, and Figure 6-2: phy LCLK *was* phy PCLK.

- IEEE Std 1394-1995 and 1394b Compliant
- Support industry standard FireWire 800 PHY
- Support Asynchronous Transfers at 100, 200, 400 and 800 Mbits/s
- Perform 1394b Cycle Master
- Implements SBP3/SBP2 stack to optimize the performance
- SBP-3 faststart support
- Shadow RAM for fast code fetch
- Programmable wait state for CPU to access registers, external SRAM, and Flash
- ATA/ATAPI-7 d1532r2, vol. 1 & 2 compliant
- Support ATA DMA modes 0-2, and UDMA 133/100/66/33
  (Note that mode 6, 133 MBytes/s, is only supported when running 1394b mode)
- Support ATA PIO mode 0-4
- Integrated internal ARM7TDMI 32-bit CPU with embedded SRAM
- Implement the firmware download mechanism
- Power Management
- Low power CMOS operating at 3.3 and 2.5 Volts
- 144-pin LQFP or 144-pin TFBGA packages available
- Dual ATA Channels
- 100 MBytes/s raw data rate read and write
- 6 GPIO pins available with TFBGA package version
- 4K payload size
- Supports up to 32 MB of external SRAM and 32 MB of Flash memory
- 16 KB of internal SRAM

Copyright

Copyright© 2003 Initio Corporation. All rights reserved. No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any, electronic, mechanical, photocopying, recording without the express written consent and authorization of Initio Corporation, 650 North Mary Ave., Sunnyvale, CA 94085

Trademarks

Initio, Inic-, are registered trademarks of Initio Corporation.

Change

Initio Corporation reserves the right to make changes in the product design without reservation and without notification to its users. The Material in this publication is for information only and is subject to change without notice.

Table of Contents

SECTION 1
Overview

1.1 Introduction

The INIC-2430 provides an advanced solution to connect ATAPI or ATA (IDE/EIDE) devices to an IEEE-1394b interface with an integrated 32-bit CPU and embedded SRAM. To provide high performance and a cost effective solution, the INIC-2430 integrates a 1394b link core, ATA control block and microprocessor into a single ASIC. The INIC-2430 delivers a data transfer rate of up to 800 Mbits/sec on FireWire 800 (1394) interface (100 MBytes/sec), while its ATA interface supports ultra DMA modes (33/66/100/133 MBytes/sec).

1.1.1 Feature Summary

- IEEE Std 1394-1995 and 1394b Compliant
- Support industry standard FireWire 800 PHY
- Support Asynchronous Transfers at 100, 200, 400 and 800 Mbits/s
- Perform 1394b Cycle Master
- Implements SBP3/SBP2 stack to optimize the performance
- SBP3/SBP2 faststart support
- Shadow RAM for fast code fetch
- Programmable wait state for CPU to access registers, external SRAM, and Flash
- ATA/ATAPI-7 d1532r2, vol. 1 & 2 compliant
- Support ATA DMA modes 0-2, and UDMA 133/100/66/33
  (Note that mode 6, 133 MBytes/s, is only supported when running 1394b mode)
- Support ATA PIO mode 0-4
- Integrated internal ARM7TDMI 32-bit CPU with embedded SRAM
- Implement the firmware download mechanism
- Power Management
- Low power CMOS operating at 3.3 and 2.5 Volts
- 144-pin LQFP or 144-pin TFBGA packages available
- Dual ATA Channels
- 100 MBytes/s raw data rate read and write
- 6 GPIO pins available with TFBGA package version
- 4K payload size
- Supports up to 32 MB of external SRAM and 32 MB of Flash memory
- 16 KB of internal SRAM

1.1.2 Firmware/Software Support

- Protocols supported include SBP-2, SBP-3, RBC and ATA/ATAPI -7
- Software utilities for downloading the upgraded firmware code

1.1.3 Devices Support

- Hard disk drives
- CD-RW devices
- DVDs
- Removable media devices
- Tape devices

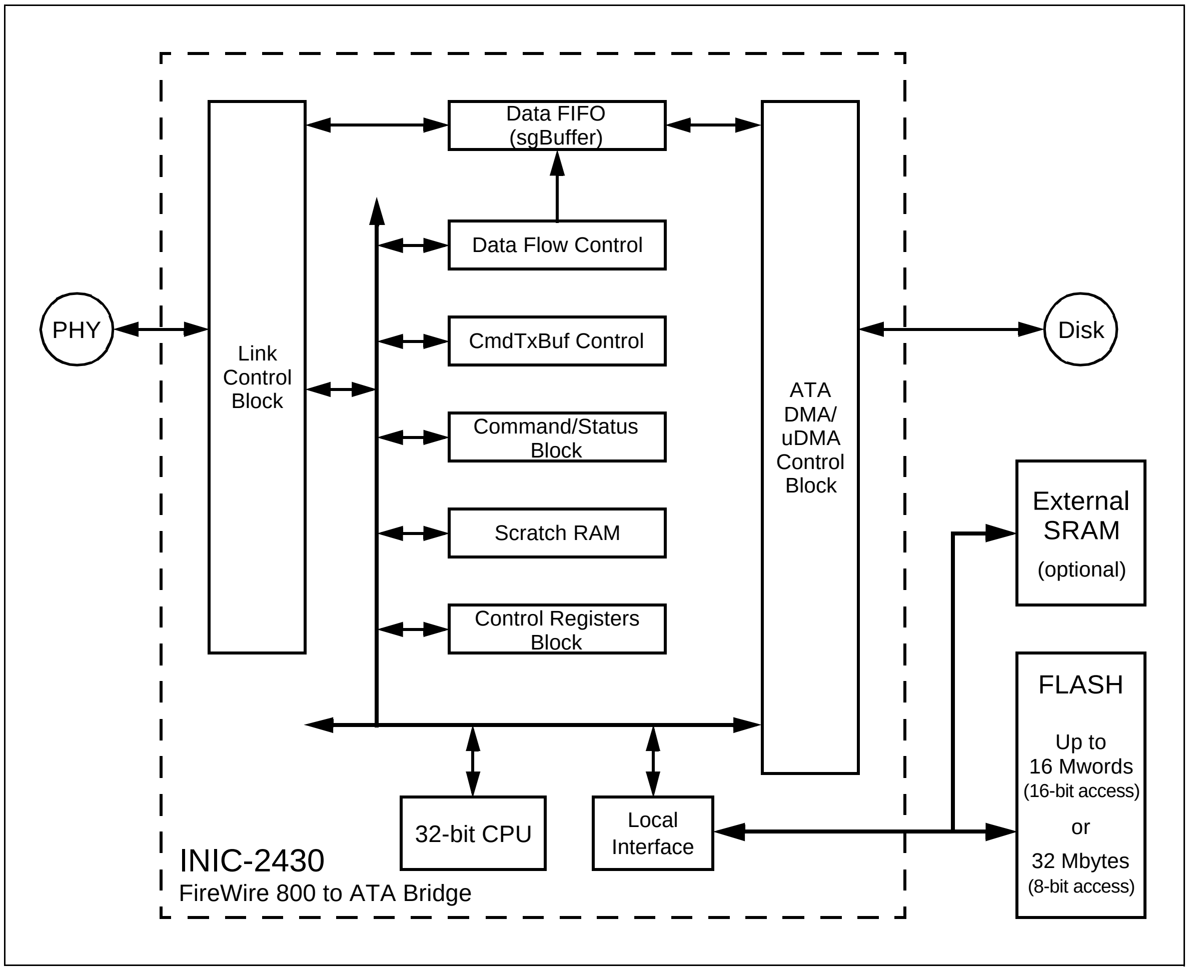


Figure 1-1 INIC-2430 Block Diagram

1.1.4 Reference Documents

- IEEE Std 1394-1995 Specification
- IEEE Std 1394a-2000 and IEEE Std 1394a-2002
- IEEE Std P1212
- IEEE Std 1394b

Figure 2-1 shows the pinout of the INIC-2430.

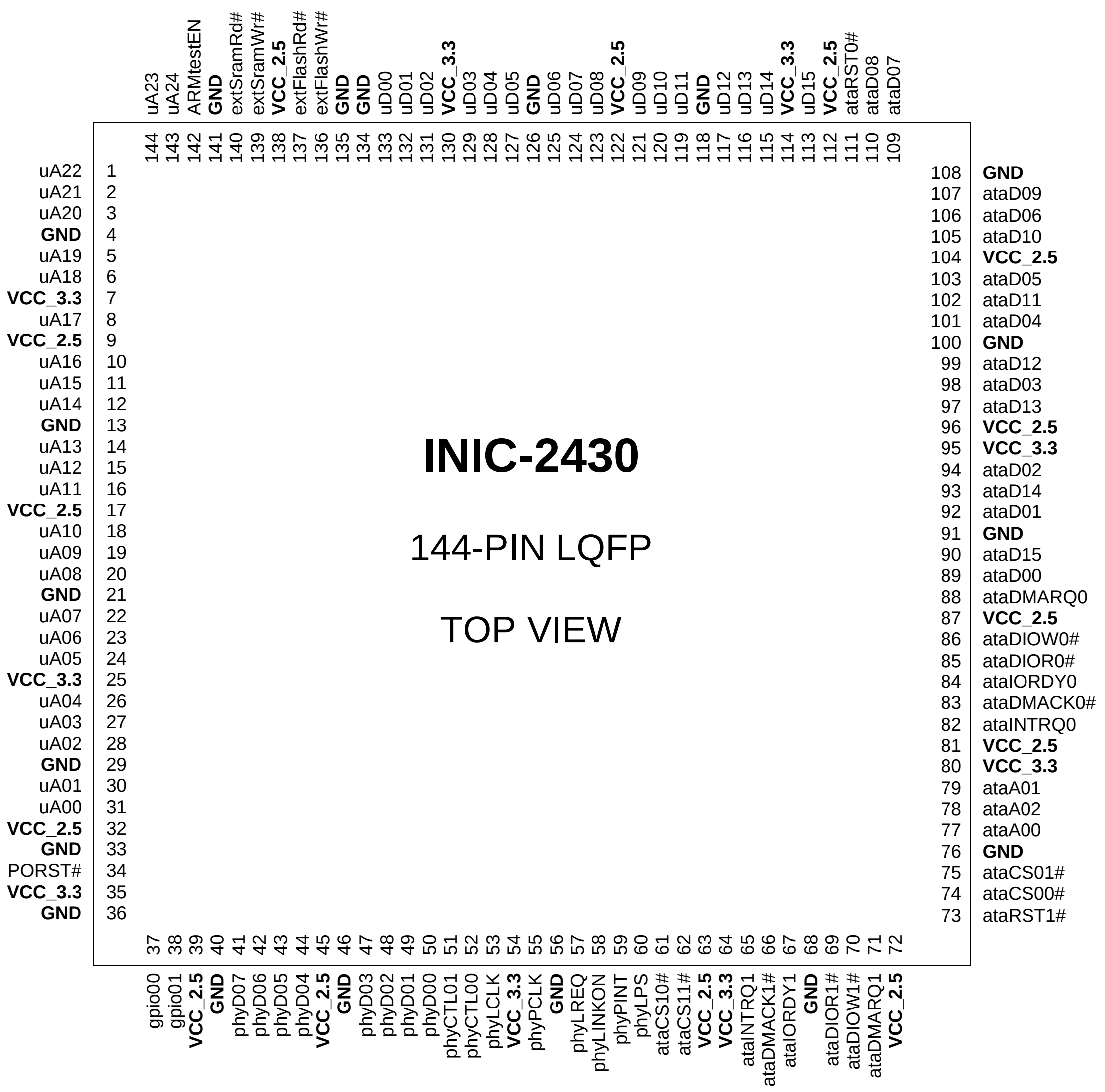


Figure 2-1 INIC-2430 LQFP Pin Assignments

Figure 2-2 shows the pinout of the INIC-2430 for the TFBGA package.

A1 uA16	A2 uA17	A3 exFlashRd#	A4 uD06	A5 uD04	A6 uD02	A7 uD00	A8 ataCS11#	A9 ataINTRQ1	A10 ataCS00#	A11 ataCS01#	A12 ataA00
B1 uA14	B2 uA15	B3 exFlashWr#	B4 uD07	B5 uD05	B6 uD03	B7 uD01	B8 ataCS10#	B9 ataDMACK1#	B10 ataDIOR1#	B11 ataA02	B12 ataA01
C1 uA12	C2 uA13	C3 uD13	C4 uD12	C5 uD11	C6 uD10	C7 uD09	C8 uD08	C9 ataIORDY1	C10 ataDIOW1#	C11 ataD09	C12 ataD06
D1 uA10	D2 uA11	D3 uD14	D4 VCC_2.5	D5 VCC_2.5	D6 uMAS1	D7 ext_test	D8 VCC_3.3	D9 VCC_3.3	D10 ataDMARQ1	D11 ataD10	D12 ataD04
E1 uA08	E2 uA09	E3 uD15	E4 VCC_2.5	E5 GND	E6 GND	E7 GND	E8 GND	E9 VCC_3.3	E10 ataRST1#	E11 ataD03	E12 ataD12
F1 uA06	F2 uA07	F3 exSramWE#	F4 VCC_2.5	F5 GND	F6 GND	F7 GND	F8 GND	F9 VCC_3.3	F10 ataD00	F11 ataD15	F12 ataD14
G1 uA04	G2 uA05	G3 exSramRE#	G4 VCC_3.3	G5 GND	G6 GND	G7 GND	G8 GND	G9 VCC_2.5	G10 ataD01	G11 ataD02	G12 ataD13
H1 uA02	H2 uA03	H3 ARMtestEN	H4 VCC_3.3	H5 GND	H6 GND	H7 GND	H8 GND	H9 VCC_2.5	H10 ataDMARQ0	H11 ataDIOW0#	H12 ataD11
J1 uA00	J2 uA01	J3 uA20	J4 VCC_3.3	J5 VCC_3.3	J6 uRW#	J7 uWAIT#	J8 VCC_2.5	J9 VCC_2.5	J10 ataDIOR0#	J11 ataD05	J12 ataIORDY0
K1 uA18	K2 uA19	K3 uA21	K4 uA22	K5 gpio03	K6 gpio04	K7 gpio05	K8 uMREQ#	K9 uMAS0	K10 ataINTRQ0	K11 ataDMACK0#	K12 ataD08
L1 PORST#	L2 uA23	L3 uA24	L4 gpio00	L5 gpio01	L6 gpio02	L7 phyLPS	L8 phyLinkOn	L9 phyPINT	L10 ataD07	L11 ataRST0#	L12 phyLREQ
M1 phyD07	M2 phyD06	M3 phyD05	M4 phyD04-	M5 phyD03	M6 phyD02	M7 phyD01	M8 phyD00	M9 phyCTL01	M10 phyCTL00	M11 phyLCLK	M12 phyPCLK

Top View

Figure 2-2 INIC-2430 TFBGA Pin Assignments

2.1 LINK-PHY Interface Pins

Table 2-1 LINK-PHY Interface Pins

SYMBOL	LQFP PIN #	TFBGA PIN #	TYPE	DESCRIPTION
phyPCLK	55	M12	I	PHY PCLOCK: Clock from PHY.
phyLCLK	53	M11	O 4 mA	PHY LCLOCK: Clock to PHY.
phyPINT	59	L9	I	PHY INTERRUPT: Interrupt from PHY to the Link.
phyLPS	60	L7	O 4 mA	PHY LINK PS: Link Power Status.
phyLINKON	58	L8	I pull up	PHY LINK ON: LinkOn from the PHY to the Link. Pull up resistance is 40-190 KΩ, 75 KΩ typical.
phyLREQ	57	L12	O 4 mA	PHY LINK REQUEST: Link Request signal.
phyD[7:0]	41-44, 47-50	M1-M8	I/O 4 mA	PHY DATA: Data Bus between Link and PHY.
phyCTL[1:0]	51, 52	M9, M10	I/O 4 mA	PHY CONTROL: Control Bus between Link and PHY.

2.2 ATA Interface Pins

ATA interface pins are for 3.3 volt logic levels but are 5 volt tolerant.

∗ NOTE: All ATA Interface pin outputs are 4 to 10 mA, programmable.

Table 2-2 ATA Interface Pins

SYMBOL	LQFP PIN #	TFBGA PIN #	TYPE	DESCRIPTION
ataD[15:0]	90, 93, 97, 99, 102, 105, 107, 110, 109, 106, 103, 101, 98, 94, 92, 89	F11, F12, G12, E12, H12, D11, C11, K12, L10, C12, J11, D12, E11, G11, G10, F10	I/O 4 mA*	ATA DATA [15:0]: ATA data bus.
ataCS00#	74	A10	O 4 mA*	ATA DEVICE CHIP SELECT 00: This signal is active low.
ataCS01#	75	A11	O 4 mA*	ATA DEVICE CHIP SELECT 01: This signal is active low.
ataCS10#/ tstp06/rs232txd	61	B8	O 4 mA*	ATA DEVICE CHIP SELECT 10: This signal is active low.
ataCS11#/ tstp07/rs232rxd	62	A8	I/O 4 mA*	ATA DEVICE CHIP SELECT 11: This signal is active low.
ataDMARQ0	88	H10	I/O 4 mA*	ATA DMA REQUEST 0
ataDMARQ1/ nTRST/tstp00/ GPIO[5]	71	D10	I/O 4 mA*	ATA DMA REQUEST 1: The GPIO[5] function is brought out to a separate pin in the TFBGA package.
ataDMACK0#	83	K11	I/O 4 mA*	ATA DMA ACKNOWLEDGE 0: This signal is active low.
ataDMACK1#/ TCK/tstp01	66	B9	I/O 4 mA*	ATA DMA ACKNOWLEDGE 1: This signal is active low.
ataDIOW0#	86	H11	O 4 mA*	ATA I/O WRITE 0: This signal is active low.
ataDIOW1#/ TMS/tstp04/ GPIO[4]	70	C10	I/O 4 mA*	ATA I/O WRITE 1: The GPIO[4] function is brought out to a separate pin in the TFBGA package.
ataDIOR0#	85	J10	O 4 mA*	ATA I/O READ 0: This signal is active low.

Table 2-2 ATA Interface Pins (Continued)

SYMBOL	LQFP PIN #	TFBGA PIN #	TYPE	DESCRIPTION
ataDIOR1#/ TDO/tstp03/ GPIO[3]	69	B10	I/O 4 mA*	ATA I/O READ 1: The GPIO[2] function is brought out to a separate pin in the TFBGA package.
ataIORDY0	84	J12	I	ATA I/O READY 0
ataIORDY1/ TDI/tstp02/ GPIO[2]	67	C9	I/O 4 mA*	ATA I/O READY 1: The GPIO[2] function is brought out to a separate pin in the TFBGA package.
ataINTRQ0	82	K10	I	ATA INTERRUPT REQUEST 0
ataINTRQ1	65	A9	I	ATA INTERRUPT REQUEST 1
ataA[2:0]	78, 79, 77	B11, B12, A12	O 4 mA*	ATA DEVICE ADDRESS [2:0]
ataRST0#	111	L11	O 4 mA*	ATA RESET 0: This bit is active low.
ataRST1#/ tstp05/Flash16	73	E10	I/O 4 mA*	ATA RESET 1: PowerOn strap to select FLASH type: 0: 8-bit FLASH 1: 16-bit FLASH

2.3 MVRAM / GPIO Interface Pins

Table 2-3 MVRAM/GPIO Interface Pins

SYMBOL	LQFP PIN #	TFBGA PIN #	TYPE	DESCRIPTION
GPIO[1:0]	38, 37	L5, L4	I/O 4 mA	GENERAL PURPOSE I/O [1:0]
GPIO[5:2]	(multiplexed with other functions in LQFP pkg)	K7-K5, L6	I/O 4 mA	GENERAL PURPOSE I/O [5:2]: These pins are multiplexed with other functions in the LQFP package and brought out separately (as shown here) in the TFBGA package.

2.4 Local Microprocessor Interface Pins

Table 2-4 Local Microprocessor Interface Pins

SYMBOL	LQFP PIN #	TFBGA PIN #	TYPE	DESCRIPTION
uD[15:0]	113, 115-117, 119-121, 123-125, 127-129, 131-133	E3, D3, C3-C8, B4, A4, B5, A5, B6, A6, B7, A7	I/O 8 mA	LOCAL DATA BUS [15:0]
uA[24]	143	L3	O 8 mA	LOCAL ADDRESS BUS [24]
uA[23] / A1394	144	L2	I/O 8 mA	LOCAL ADDRESS BUS [23]: Output - Local Address Bus bit 23. Input - 0: 1394B Mode, 1: 1394A Mode. (This pin should be tied high or low with a pull up/down resistor for Power-On Reset.)
uA[22:0]	1-3, 5, 6, 8, 10-12, 14-16, 18-20, 22-24, 26-28, 30, 31	K4, K3, J3, K2, K1, A2, A1, B2, B1, C2, C1, D2, D1, E2, E1, F2, F1, G2, G1, H2, H1, J2, J1	O 8 mA	LOCAL ADDRESS BUS [22:0]
uMAS[1:0]	n/a	D6, K9	I	LOCAL MAS [1:0]: These pins are for manufacturing use **only** and must be tied low for normal operation.
uMREQ#	n/a	K8	I	LOCAL MASTER REQUEST: This pin is for manufacturing use **only** and must be tied low for normal operation.
uRW#	n/a	J6	I	LOCAL R/W: This pin is for manufactur-ing use **only** and must be tied low for nor-mal operation.
uWAIT#	n/a	J7	I	LOCAL WAIT: This pin is for manufactur-ing use **only** and must be tied low for nor-mal operation.
ext_test	n/a	D7	I	EXTERNAL TEST: This pin is for manu-facturing use **only** and must be tied low for normal operation.
extSramRd#	140	G3	O 8 mA	EXTERNAL SRAM READ STROBE: This bit is active low.
extSramWr#	139	F3	O 8 mA	EXTERNAL SRAM WRITE STROBE: This bit is active low.

Table 2-4 Local Microprocessor Interface Pins (Continued)

SYMBOL	LQFP PIN #	TFBGA PIN #	TYPE	DESCRIPTION
extFlashRd#	137	A3	O 8 mA	LOCAL EXTERNAL FLASH READ STROBE: This bit is active low.
extFlashWr#	136	B3	O 8 mA	LOCAL EXTERNAL FLASH WRITE STROBE: This bit is active low.

2.5 System Interface Pins

Table 2-5 System Interface Pins

SYMBOL	LQFP #	TFBGA #	TYPE	DESCRIPTION
PORST#	34	L1	I	POWER ON RESET: Active low. When this signal is active, all pins on the ATA interface should be tri-stated.
ARMtestEN	142	H3	I	ARM TEST ENABLE: 0: Normal, 1: Test internal ARM.

2.6 Power and Ground Pins

Table 2-6 Power and Ground Pins

SYMBOL	LQFP PIN #	TFBGA PIN #	TYPE	DESCRIPTION
VCC_2.5	9, 17, 32, 39, 45, 63, 72, 81, 87, 96, 104, 112, 122, 138	D4, D5, E4, F4, G9, H9, J8, J9	PWR (2.5V)	POWER for internal core logic.
VCC_3.3	7, 25, 35, 54, 64, 80, 95, 114, 130	D8, D9, E9, F9, G4, H4, J4, J5	PWR (3.3V)	POWER for I/O logic.
GND	4, 13, 21, 29, 33, 36, 40, 46, 56, 68, 76, 91, 100, 108, 118, 126, 134, 135, 141	E5-E8, F5-F8, G5-G8, H5-H8	GND	GROUND for internal core logic and I/O pins.

This page is intentionally left blank.

3.1 Address Mapping

	Address Area	No Shadow (PowerOn default)	Shadow to Internal SRAM	Shadow to External SRAM
	Flash	0000_0000 | (32 Mbytes) 01FF_FFFF	C000_0000 | (32 Mbytes) C1FF_FFFF	4000_0000 | (32 Mbytes) 41FF_FFFF
	External SRAM (32 Mbytes)	4000_0000 | 41FF-FFFF		0000_0000 | 01FF-FFFF
	mirrored Flash - see Note 1 - (32 Mbytes)	8000_0000 | 81FF-FFFF		
↑	Internal SRAM (16 Kbytes)	C000_0000 | C000-3FFF	0000_0000 | 0000-3FFF	C000_0000 | C000-3FFF
|	*Reserved*	C000_4000 | (1/2 Gbytes - 16 Kbytes) DFFF_FFFF	0000_4000 | (1 Gbyte - 16 Kbytes) 3FFF_FFFF	C000_4000 | (1/2 Gbytes - 16 Kbytes) DFFF_FFFF
1 Gbyte	Registers (512 Bytes)	E000_0000 | E000_01FF		
|	Buffers (3584 Bytes)	E000_0200 | E000_0FFF		
↓	*Reserved*	E000_1000 | FFFF_FFFF		

Figure 3-1 INIC-2430 Address Mapping

NOTE: This address area is mirrored to the Flash code independent of the shadow scheme.

3.1.1 Firmware Shadow Procedure

Method A: Use internal SRAM as shadow RAM

1 After power-on, the Firmware will start fetching code from the Flash (code space 0000_0000), and copy the code from the Flash (data space 0000_0000 — 01FF_FFFF) into the internal SRAM (data space C000_0000 — C000_3FFF).

2 When the copy is done, the firmware should set the register bit *shadowEn* (reg. E000_00BAh, bit 7) to 1, which will enable the shadowed SRAM. Another register bit, *shadowExt* (reg. E000_00BAh, bit 6), needs to be set to 0, which will select internal SRAM as the shadow RAM.

3 With *shadowEn* = 1 & *shadowExt* = 0, the code fetch range 0000_0000 — 3FFF_FFFF will be from internal SRAM, and

4 The Flash will be re-located to code fetch range C000_0000 — C1FF_FFFF.

Method B: Use external SRAM as shadow RAM

1 After power-on, the Firmware will start fetching code from the Flash (code space 0000_0000), and copy the code from the Flash (data space 0000_0000 — 01FF_FFFF) into the external SRAM (data space 4000_0000 — 41FF_FFFF).

2 When the copy is done, the firmware should set the register bit *shadowEn* (reg. E000_00BAh, bit 7) to 1, which will enable the shadowed SRAM. Another register bit, *shadowExt* (reg. E000_00BAh, bit 6), also needs to be set to 1, which will select external SRAM as the shadow RAM.

3 With *shadowEn* = 1 & *shadowExt* = 1, the code fetch range 0000_0000 — 3FFF_FFFF will be from external SRAM, and

4 The Flash will be re-located to code fetch range 4000_0000 — 41FF_FFFF.

NOTE: To make firmware programming easier to access FLASH:

With Register E000_0121 bit 0 = 1’b0 (default):

Address range 8000-81FF is hardcoded for FLASH access (not affected by shadow scheme).

3.2 Registers

NOTE: CPU accesses the internal registers:

E000_0000h - E000_007Fh: use 16/32-bit mode (Link registers).

E000_0090h: use 16-bit mode (ATA data register).

E000_0091h - E000_009Fh: use 8/16-bit mode (ATA control registers).

E000_00A0h - E000_00A7h: use 16/32-bit mode (data port to WR/RD S/G FIFOs).

E000_00B0h - E000_0153h: use 8/16/32-bit mode (Bridge general registers).

NOTE: CPU will access the ATA device registers at the addresses shown in Section 3.2.2:

3.2.1 Link Block

Address	Read Value	Write Value
E000_0000h	XATRetries	XATRetries
E000_0004h	YATRetries	YATRetries
E000_0008h	ZATRetries	ZATRetries
E000_000Ch	TxAckCntrl	TxAckCntrl
E000_0010h	CycStartSpdCntrl	CycStartSpdCntrl
E000_0014h	*Reserved*	*Reserved*
E000_0018h	SelfIDCount	SelfIDCount
E000_001Ch	*Reserved*	*Reserved*
E000_0020h	ChannelRcvHi	ChannelRcvHi_Set
E000_0024h	ChannelRcvHi	ChannelRcvHi_Clr
E000_0028h	ChannelRcvLo	ChannelRcvLo_Set
E000_002Ch	ChannelRcvLo	ChannelRcvLo_Clr
E000_0030h	IntEvent	IntEvent_Set
E000_0034h	IntEvent & IntEnable	IntEvent_Clr
E000_0038h	IntEnable	IntEnable_Set
E000_003Ch	IntEnable	IntEnable_Clr
E000_0040h	FairnessCntrl	FairnessCntrl
E000_0044h	PingCntReg	PingCntReg
E000_0048h	LinkCntrl	LinkCntrl_Set
E000_004Ch	LinkCntrl	LinkCntrl_Clr
E000_0050h	NodeID	NodeID
E000_0054h	PhyCntrl	PhyCntrl
E000_0058h-E000_007Ch	*Reserved*	*Reserved*

3.2.2 ATA Block

NOTE: The ATA device registers are part of an external ATA device. These registers are accessed at the addresses shown below:

Address	Read Value	Write Value
E000_0090h	Data[15:0] (16-bit access)	Data[15:0] (16-bit access)
E000_0091h	Error	Features
E000_0092h	SectorCount	SectorCount
E000_0093h	SectorNumber	SectorNumber
E000_0094h	CyclinderLow	CylinderLow
E000_0095h	CylinderHigh	CylinderHigh
E000_0096h	Device/Head	Device/Head
E000_0097h	Status	Command
E000_0098h-E000_009Dh	*Reserved*	*Reserved*
E000_009Eh	AlternateStatus	DeviceControl
E000_009Fh	*Reserved*	*Reserved*

3.2.3 Data Port to WR/RD S/G FIFOs

Address	Read Value	Write Value
E000_00A0h	FIFO0D[7:0] (32-bit access)	FIFO0D[7:0] (32-bit access)
E000_00A1h	FIFO0D[15:8]	FIFO0D[15:8]
E000_00A2h	FIFO0D[23:16]	FIFO0D[23:16]
E000_00A3h	FIFO0D[31:24]	FIFO0D[31:24]
E000_00A4h	FIFO1D[7:0] (32-bit access)	FIFO1D[7:0] (32-bit access)
E000_00A5h	FIFO1D[15:8]	FIFO1D[15:8]
E000_00A6h	FIFO1D[23:16]	FIFO1D[23:16]
E000_00A7h	FIFO1D[31:24]	FIFO1D[31:24]

3.2.4 Bridge General Registers

Address	Read Value	Write Value
E000_00B0h	sgPCtrl	sgPCtrl
E000_00B1h	FifoSts	FifoSts
E000_00B2h	gpioData	gpioData
E000_00B3h	gpioCtrl	gpioCtrl
E000_00B4h	TestCtrl0	TestCtrl0
E000_00B5h	TestCtrl1	TestCtrl1
E000_00B6h	DrvCtrl	DrvCtrl

Address	Read Value	Write Value
E000_00B7h	Agnet0Stat	Agent0Stat
E000_00B8h	Agent1Stat	Agent1Stat
E000_00B9h	Agent1Ofs	Agent1Ofs
E000_00BAh	uPCtrl	uPCtrl
E000_00BBh	PwrMgntCtrl	PwrMgntCtrl
E000_00BCh	RevID	N/A
E000_00C0h	LinkCtrl	LinkCtrl
E000_00C1h	DmaCtrl	DmaCtrl
E000_00C2h-E000_00C3h	*Reserved*	*Reserved*
E000_00C4h	AtaCtrl (PIO/DMA EN)	AtaCtrl (PIO/DMA EN)
E000_00C5h	AtaMstrCtrl (PIO/DMA mode for Master Device)	AtaMstrCtrl (PIO/DMA mode for Master Device)
E000_00C6h	AtaSlvCtrl (PIO/DMA mode for Slave Device)	AtaSlvCtrl (PIO/DMA mode for Slave Device)
E000_00C7h	AtaStatus	AtaStatus
E000_00C8h-E000_00CFh	*Reserved*	*Reserved*
E000_00D0h	LoginID0[7:0] (16-bit access)	LoginID0[7:0] (16-bit access)
E000_00D1h	LoginID0[15:8]	LoginID0[15:8]
E000_00D2h	LoginID1[7:0] (16-bit access)	LoginID1[7:0] (16-bit access)
E000_00D3h	LoginID1[15:8]	LoginID1[15:8]
E000_00E0h	Ata_wr_sg_threshold	Ata_wr_sg_threshold
E000_00E1h	*Reserved*	*Reserved*
E000_00E2h	Faststart0/1_offset	Faststart0/1_offset
E000_00E3h-E000_00E4h	*Reserved*	*Reserved*
E000_00E5h	Baud rate select	Baud rate select
E000_00E6h	start/stop/parity bit select	start/stop/parity bit select
E000_00E7h	N/A	RS232 WrPort
E000_00E8h	RS232 WrPort Status	N/A
E000_00E9h	RS232 RdPort	N/A
E000_00EAh	RS232 RdPort Status	N/A
E000_00EBh	Timer_current_cnt[7:0]	Timer_count[7:0]
E000_00ECh	Timer_current_cnt[15:8]	Timer_count[15:8]
E000_00EDh	Timer Timeout bit, Timeout_to_INT_Enable bit	Timer Timeout bit, Timeout_to_INT_Enable bit
E000_00F0h	Wait_state_ARM_rw_Reg	Wait_state_ARM_rw_Reg
E000_00F1h	Wait_state_ARM_rw_SRAM_ext	Wait_state_ARM_rw_SRAM_ext
E000_00F2h	Wait_state_ARM_rw_Flash	Wait_state_ARM_rw_Flash
E000_00F3h	Wait_state_ARM_rw_SRAM_int	Wait_state_ARM_rw_SRAM_int
E000_00F4h	Wait_state_ARM_rw_Buffer	Wait_state_ARM_rw_Buffer
E000_00F5h	Wait_state_ARM_rw_Link_Reg	Wait_state_ARM_rw_Link_Reg
E000_00F6h-E000_00FFh	*Reserved*	*Reserved*
E000_0100h-010Fh	*Reserved*	*Reserved*

Address	Read Value	Write Value
E000_0110h	SgRun (32-bit access)	sgRun_Set
E000_0111h	sgRun	sgRun_Clr
E000_0112h	sgError	sgError_Clr
E000_0113h	sgRetryExcd status	sgRetryExcd clear
E000_0114h-E000_0117h	*Reserved*	*Reserved*
E000_0118h	CmdTxRun (8/16/32-bit access)	cmdTxRun_Set (8/16/32-bit access)
E000_0119h	CmdTxRun (8-bit access)	cmdTxRun_Clr (8-bit access)
E000_011Ah	CmdTxError (8/16-bit access)	cmdTxError_Clr (8/16-bit access)
E000_011Bh	CmdTxRetryExcd (8-bit access)	cmdTxRetryExcd_Clr (8-bit access)
E000_011Ch	CmdTxReset (same as RD Register 011Ah)	CmdTx state machine resume
E000_011Dh	cmdtx2/0_busy status (RD)	WR 1 to clear cmdtx2/0_busy (WR)
E000_011Eh	CmdrxActiveStatus	CmdrxActiveStatus_Clr
E000_0120h	AckRetry_En	AckRetry_En
E000_0121h	rCodeRetry_En	rCodeRetry_En
E000_0122h	ReReq_Ch_En	ReReq_Ch_En
E000_0123h-E000_012Fh	*Reserved*	*Reserved*
E000_0130h	tCode_rCode	tCode_rCode
E000_0131h-E000_013Fh	*Reserved*	*Reserved*
E000_0140h	INT0_Status (8/16/32-bit access)	INT0_Clear (8/16/32-bit access)
E000_0141h	INT1_Status (8-bit access)	INT1_Clear (8-bit access)
E000_0142h	INT2_Status (8/16-bit access)	INT2_Clear (8/16-bit access)
E000_0143h	INT3_Status (8-bit access)	INT3_Clear (8-bit access)
0144h-014Fh	*Reserved*	*Reserved*
E000_0150h	INT0_En (8/16/32-bit access)	INT0_En (8/16/32-bit access)
E000_0151h	INT1_En (8-bit access)	INT1_En (8-bit access)
E000_0152h	INT2_En (8/16-bit access)	INT2_En (8/16-bit access)
E000_0153h	INT3_En (8-bit access)	INT3_En (8-bit access)
E000_0154h	Trailer0_byte0	NA
E000_0155h	Trailer0_byte1	NA
E000_0156h	Trailer0_byte2	NA
E000_0157h	Trailer0_byte3	NA
E000_0158h	Trailer1_byte0	NA
E000_0159h	Trailer1_byte1	NA
E000_015Ah	Trailer1_byte2	NA
E000_015Bh	Trailer1_byte3	NA
E000_015Ch	MiscEn	MiscEn (internal testing)
E000_0160h-E000_016F	*Reserved*	*Reserved*
E000_0170h	NA	Split_Timer_L
E000_0171h	NA	Split_Timer_H

3.3 Buffers

NOTE: CPU accesses the internal buffers:

E000_0200h - E000_02BFh: cmdRxBuffers: RD: use 8/16/32-bit mode.
(CPU is not allowed to write cmdRxBuffers)

E000_0300h - E000_0DFFh: cmdTxBuffers: WR/RD: use 8/16/32-bit mode.

E000_0E00h - E000_0E7Fh: S/G List0-3: 8/16/32-bit mode.

3.3.1 CMD/DATA Block

Address	Read Value	Write Value
E000_0200h - E000_023Fh	CmdRx0Buffer (64 bytes)	Cannot be written by CPU
E000_0280h - E000_02BFh	CmdRx1Buffer (64 bytes)	Cannot be written by CPU
E000_0300h - E000_03FFh	CmdTx0Buffer (256 bytes)	CmdTx0Buffer (256 bytes)
E000_0500h - E000_053Fh	CmdTx1Buffer (64 bytes)	CmdTx1Buffer (64 bytes)
E000_0600h - E000_06FFh	CmdTx2Buffer (256 bytes)	CmdTx2Buffer (256 bytes)
E000_0800h - E000_083Fh	CmdTx3Buffer (64 bytes)	CmdTx3Buffer (64 bytes)
E000_0900h - E000_091Fh	CmdTx4Buffer's Header (20 bytes)	CmdTx4Buffer's Header (20 bytes)
E000_0A00h - E000_0A7Fh	CmdTx4Buffer's Data (128 bytes)	CmdTx4Buffer's Data (128 bytes)
E000_0C00h - E000_0C1Fh	CmdTx5Buffer's Header (20 bytes)	CmdTx5Buffer's Header (20 bytes)
E000_0D00h - E000_0D7Fh	CmdTx5Buffer's Data (128 bytes)	CmdTx5Buffer's Data (128 bytes)

NOTE: **For outgoing packets**, the local link module expects a 5 quadlet-header from cmdtxBuffer or sgList, and will process this 5-quadlets header, and send out a 4-quadlets header to Firewire 800 bus (complies with 1394 protocol).

For incoming packets, the header always has 4 quadlets.

3.3.1.1 Asynchronous Receive Packet Formats

The following are the header formats used with Asynchronous Receive Packets. The LINK_CORE accepts only receive packets with valid "tCode" as defined by IEEE Std 1394a-2000; packets with undefined "tCode" are rejected. Refer to Table 3-1 for descriptions of the fields used.

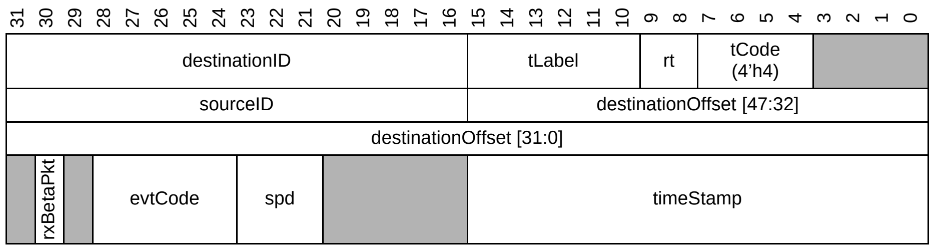


Figure 3-2 RxRdReqDQ - Quadlet Read Request Receive Packet Format

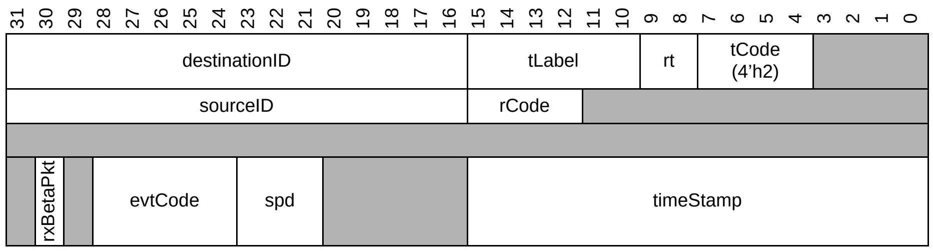


Figure 3-3 RxWrResp - Write Response Receive Packet Format

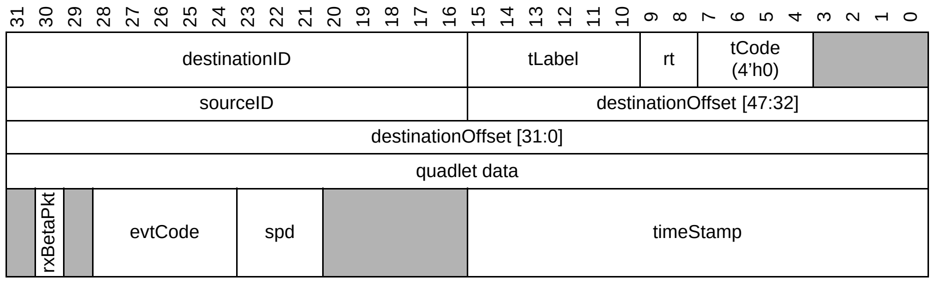


Figure 3-4 RxWrReqDQ - Quadlet Write Request Receive Packet Format

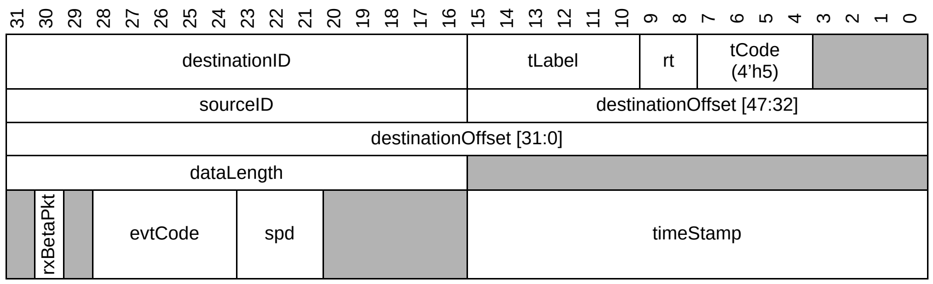


Figure 3-5 RxRdReqDB - Block Read Request Receive Packet Format

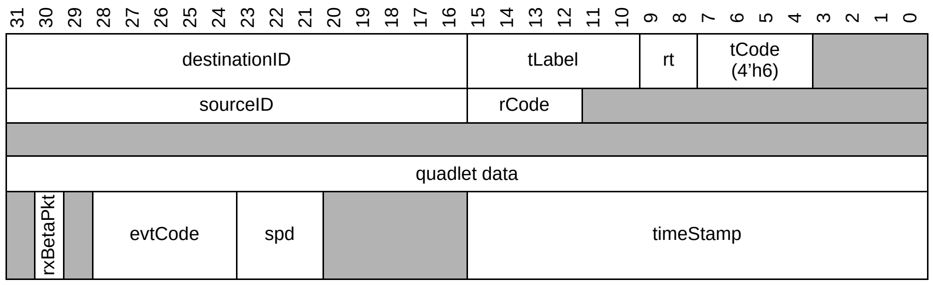


Figure 3-6 RxRdRespDQ - Quadlet Read Response Receive Packet Format

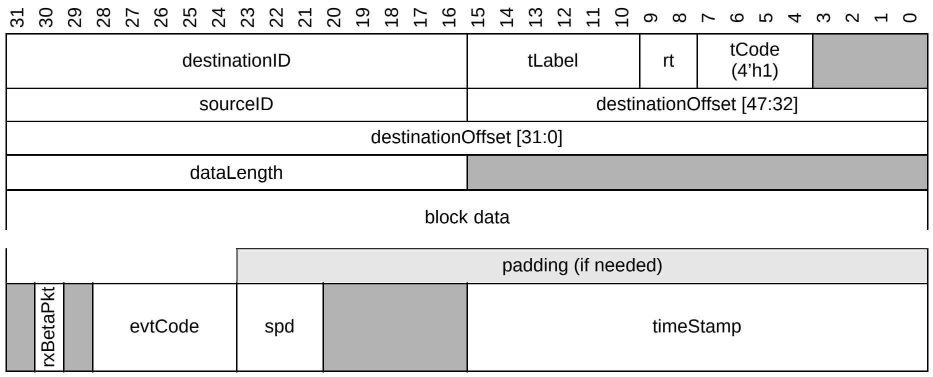


Figure 3-7 RxWrReqDQ - Block Write Request Receive Packet Format

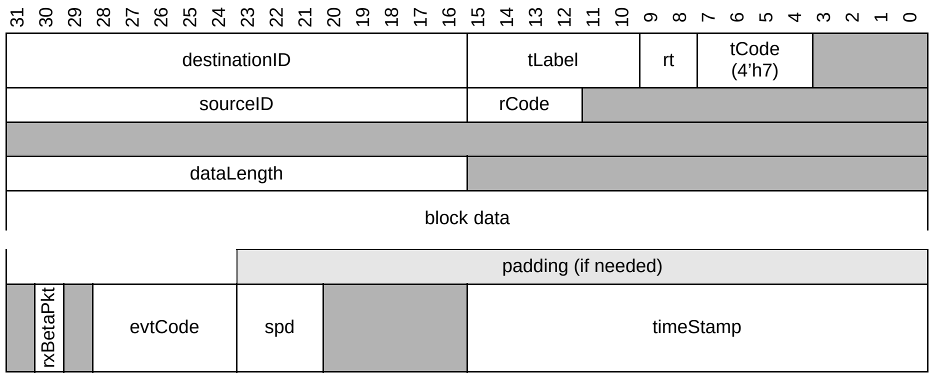


Figure 3-8 RxRdRespDB - Block Read Response Receive Packet Format

Table 3-1 Asynchronous Receive Packet Format Field Definitions

Field Name	Bit(s)	Packet Type	Description
1st Quadlet			
destinationID	31:16	all packets	DESTINATION ID: This is the concatenation of the 10-bit bus number and the 6-bit node number for the Destination of this packet. This field gives the destination ID of the node to which the packet was transmitted.
tLabel	15:10	all packets	TRANSACTION LABEL: This field is used to pair up a response packet with its corresponding request packet.
rt	9:8	all packets	RETRY CODE: This is the retry code for this packet. The LINK_CORE ignores the software provided retry code and substitutes an rt as appropriate for the implemented retry mechanism unless the rtsel bit is set to 1.
tCode	7:4	all packets	TRANSACTION CODE: This is the transaction code for this packet.
2nd & 3rd Quadlets			
sourceID	31:16	all packets	SOURCE ID: Node-ID (bus-ID + physical-ID of the sender of this packet.
destinationOffset [47:32] destinationOffset [31:0]	15:0 (2nd) 31:0 (3rd)	RxRdReqDQ RxWrReqDQ RxRdReqDB RxWrReqDB	DESTINATION OFFSET: The concatenation of these two fields addresses a quadlet in the destination node's Address space. For block data payload packets, the concatenation of these two fields indicates the starting address of block data payload. This address must be quadlet-aligned (modulo 4).
rCode	15:12	RxWrResp RxRdRespDQ RxRdRespDQ	RESPONSE CODE: This is the response code field.
4th Quadlet			
dataLength	31:16	RxWrReqDB RxRdRespDB RxRdReqDB	DATA LENGTH: The number of bytes requested in a block read request or number of bytes in a block response packet.
Extra Quadlets			
block data		RxWrReqDB RxRdRespDB	BLOCK DATA PAYLOAD: Irrespective of the destination or source node data alignment, the first byte of the block data payload must be the most significant byte of the first quadlet in the "block data" field.
padding		RxWrReqDB RxRdRespDB	PADDING: The sender of the packet pads the data block with the required number of bytes to make the block data an integral multiple of quadlets.
Last Quadlet			
rxBetaPkt	30	all packets	RECEIVE BETA PACKET: This bit when set, indicates that the packet was received in Beta format. When clear, this bit indicates that the packet was received in Legacy format.
evtCode	28:24	all packets	EVTCODE: For a non-broadcast asynchronous receive packet, the "evtCode" field carries the acknowledge code sent by the LINK_CORE prefixed by 1'b1. For a broadcast asynchronous receive packet, "evtCode" carries "ACK_COMPLETE" prefixed by 1'b1.
spd	23:21	all packets	SPEED: This field indicates the speed at which this packet is to be transmitted. 3'b000 = 100 Mbits/sec, 3'b001 = 200 Mbits/sec, 3'b010 = 400 Mbits/sec, and 3'b011 = 800 Mbits/sec. Other values are reserved.
timeStamp	15:0	all packets	TIME STAMP: This is the low order 3 bits of IsochronousCycleTimer.*cycleSeconds* and the full 13 bits of IsochronousCycleTimer.*cycleCount* at the end of packet reception.

3.3.1.2 Asynchronous Transmit Packet Formats

The following are the header formats used with Asynchronous Transmit Packets. Refer to Table 3-2 for descriptions of the fields used.

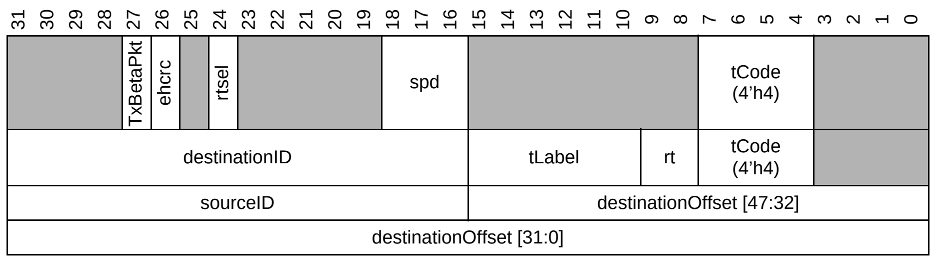


Figure 3-9 TxRdReqDQ - Quadlet Read Request Transmit Packet Format

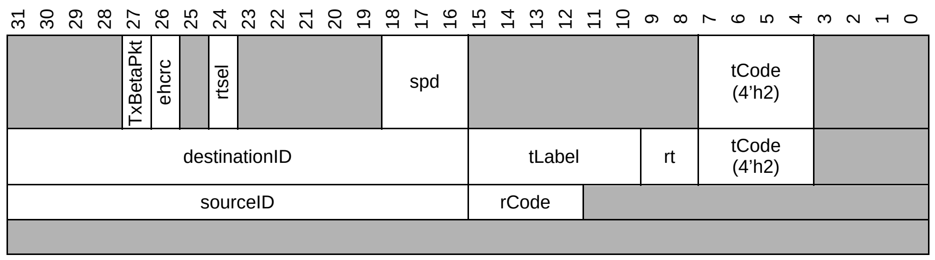


Figure 3-10 TxWrResp - Write Response Transmit Packet Format

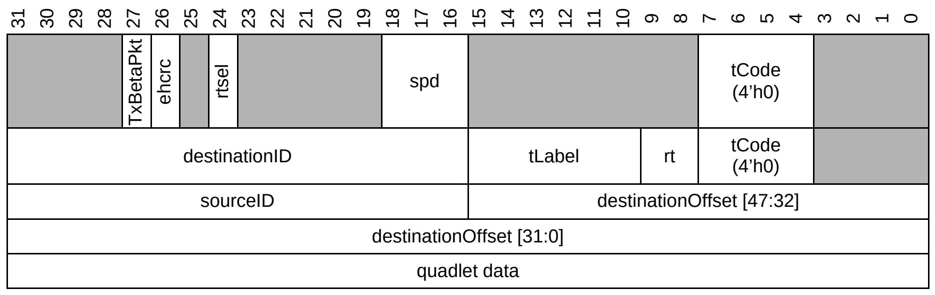


Figure 3-11 TxWrReqDQ - Quadlet Write Request Transmit Packet Format

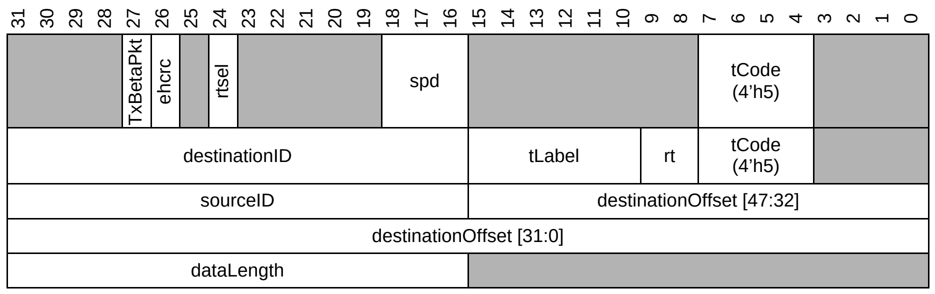


Figure 3-12 TxRdReqDB - Block Read Request Transmit Packet Format

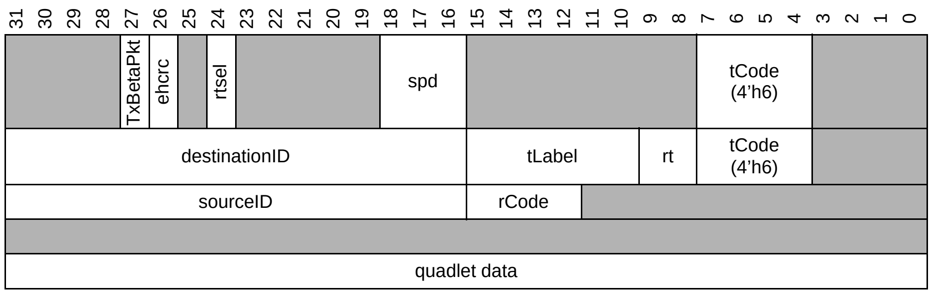


Figure 3-13 TxRdRespDQ - Quadlet Read Response Transmit Packet Format

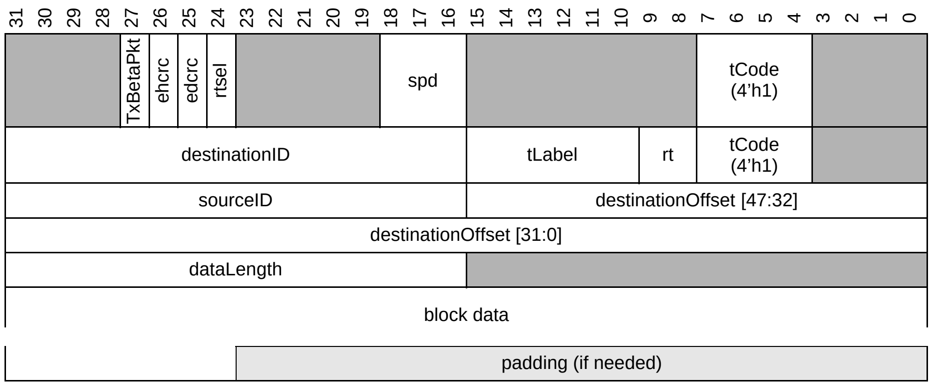


Figure 3-14 TxWrReqDB - Block Write Request Transmit Packet Format

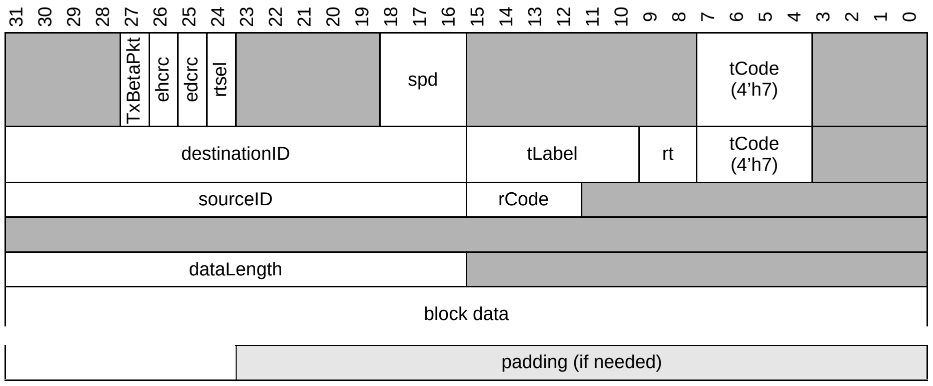


Figure 3-15 TxRdRespDB - Block Read Reponse Transmit Packet Format

Table 3-2 Asynchronous Transmit Packet Header Field Definitions

Field Name	Bit(s)	Packet Type	Description
1st Quadlet			
txBetaPkt	27	all packets	TRANSMIT BETA PACKET: This bit if set, the LINK_CORE sets the "RFMT" bit in the asynchronous packet transmit request pattern. Set this bit to explicitly request the PHY to use the Beta packet format for packet transmission.
ehcrc	26	all packets	HEADER CRC ERROR: This bit if set, the LINK_CORE sends the packet with header CRC error.
edcrc	25	TxWrReqDB TxRdRespDB	DATA CRC ERROR: This bit if set, the LINK_CORE sends the packet with data CRC error.
rtsel	24	all packets	RETRY CODE SELECT: This is the retry code select bit for this packet. The LINK_CORE ignores the software provided retry code and substitutes an rt as appropriate for the implemented retry mechanism unless the rtsel bit is set to 1.
spd	18:16	all packets	SPEED: This field indicates the speed at which this packet is to be transmitted. 3'b000 = 100 Mbits/sec, 3'b001 = 200 Mbits/sec, 3'b010 = 400 Mbits/sec, and 3'b011 = 800 Mbits/sec. Other values are reserved.
tCode	7:4	all packets	TRANSACTION CODE: This is the transaction code for this packet.
2nd Quadlet			
destinationID	31:16	all packets	DESTINATION ID: Concatenation of the 10-bit bus number and the 6-bit node number for the Destination of this packet. This field gives the destination ID of the node to which the packet was transmitted.
tLabel	15:10	all packets	TRANSACTION LABEL: This field is used to pair up a response packet with its corresponding request packet.
rt	9:8	all packets	RETRY CODE: This is the retry code for this packet. The LINK_CORE ignores the software provided retry code and substitutes an rt as appropriate for the implemented retry mechanism unless the rtsel bit is set to 1.
tCode	7:4	all packets	TRANSACTION CODE: This is the transaction code for this packet.
3rd & 4th Quadlets			
sourceID	31:16	all packets	SOURCE ID: This is the concatenation of the 10-bit bus number and the 6-bit physical ID for the transmitting node, as supplied by the host.
destinationOffset [47:32] destinationOffset [0:31]	15:0 (2nd) 31:0 (3rd)	TxRdReqDQ TxWrReqDQ TxRdReqDB TxWrReqDB	DESTINATION OFFSET: The concatenation of these two fields addresses a quadlet in the destination node's Address space. For block data payload packets, the concatenation of these two fields indicates the starting address of block data payload. This address must be quadlet-aligned (modulo 4).
rCode	15:12	TxWrResp TxRdRespDQ TxRdRespDB	RESPONSE CODE: This is the response code field.
5th Quadlet			
quadlet data	31:0	TxWrReqDQ TxRdRespDQ	QUADLET DATA: For quadlet Write Requests, this field holds the data to be written. For quadlet Read Responses, this fields holds the data quadlet read from the host address space.
dataLength	31:16	TxWrReqDB TxRdRespDB TxRdReqDB	DATA LENGTH: The number of bytes requested in a block read request or the number of bytes present in a block write request or block read response packet. If the "dataLength" field value is 16'h0, the header CRC quadlet will be the last quadlet of the packet.
Extra Quadlets			
block data		RxWrReqDB RxRdRespDB	BLOCK DATA PAYLOAD: Irrespective of the destination or source node data alignment, the first byte of the block data payload must be the most significant byte of the first quadlet in the "block data" field.
padding		RxWrReqDB RxRdRespDB	PADDING: If dataLength (*dataLength field value mod 4*) is not equal to zero, then zero-value bytes have to be added to make the packet length an integral multiple of quadlets. LINK_CORE will accept and transmit only packets that contain intergral multiple of quadlets.

3.3.1.3 PHY Receive Packet Format

The following is the header format used with PHY Receive Packets. All unspecified bits are reserved and have a value of ’0’. Refer to Table 3-3 for descriptions of the fields used.

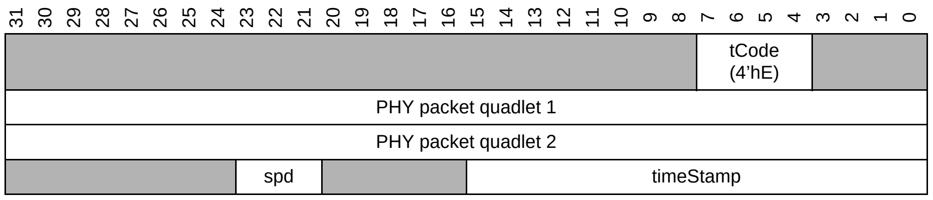


Figure 3-16 PHY Receive Packet Format

Table 3-3 PHY Receive Packet Format Field Definitions

Field Name	Bit(s)	Description
1st Quadlet		
tCode	7:4	TRANSACTION CODE: This is the transaction code for this packet. Set to 4’hE to indicate a PHY packet.
2nd Quadlet		
PHY packet (1st quadlet)	31:0	PHY PACKET 1ST QUADLET: This is the first quadlet of the received PHY packet.
3rd Quadlet		
PHY packet (2nd quadlet)	31:0	PHY PACKET 2ND QUADLET: This is the second quadlet of the received PHY packet; it is the logical inverse (ones-complement) of "PHY packet first quadlet".
4th Quadlet		
spd	23:21	SPEED: This field indicates the speed at which this packet is to be transmitted. 3’b000 = 100 Mbits/sec, 3’b001 = 200 Mbits/sec, 3’b010 = 400 Mbits/sec, and 3’b011 = 800 Mbits/sec. Other values are reserved.
timeStamp	15:0	TIME STAMP: This is the low order 3 bits of IsochronousCycleTimer.*cycleSeconds* and the full 13 bits of IsochronousCycleTimer.*cycleCount* at the end of packet reception.

3.3.1.4 Synthesized Bus Reset Packet Format

The following is the header format used with Synthesized Bus Reset Packets. All unspecified bits are reserved and have a value of ’0’. Refer to Table 3-4 for descriptions of the fields used.

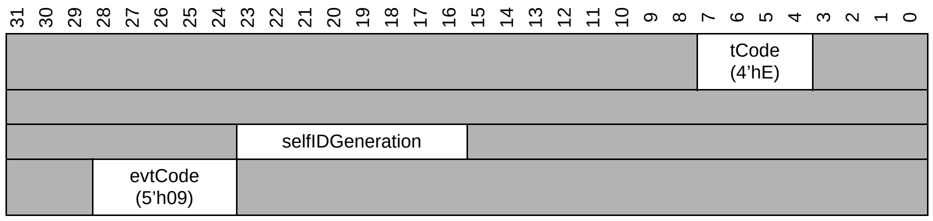


Figure 3-17 Synthesized Bus Reset Packet Format

Table 3-4 Synthesized Bus Reset Packet Format Field Definitions

Field Name	Bit(s)	Description
1st Quadlet		
tCode	7:4	TRANSACTION CODE: This is the transaction code for this packet. Set to 4’hE to indicate a PHY packet.
3rd Quadlet		
selfIDGeneration	23:16	SELF ID GENERATION: This is the SelfIDCount.selfIDGeneration value at the time this packet was created.
4th Quadlet		
evtCode	28:24	EVTCODE: A value of 5’h09 identifies it as a bus-reset packet.

3.3.2 S/G DMA Block

Address	Read Value	Write Value
E000_0E00h - E000_0E1Fh	S/G List0 (20 bytes)	S/G List0 (20 bytes)
E000_0E20h - E000_0E3Fh	S/G List1 (20 bytes)	S/G List1 (20 bytes)
E000_0E40h - E000_0E5Fh	S/G List2 (20 bytes)	S/G List2 (20 bytes)
E000_0E60h - E000_0E7Fh	S/G List3 (20 bytes)	S/G List3 (20 bytes)

NOTE: S/G List0-3 are for channel SG0-3’s headers.
S/G FIFO0-1 are ping-pong buffers for SG channel’s data.

S/G FIFOs can be accessed by CPU through the data ports:

WR access:

FIFO0: 32-bit WR to port adderss E000_00A0
or
16-bit WR to port address E000_00A0
16-bit WR to port address E000_00A2

FIFO1: 32-bit WR to port adderss E000_00A4
or
16-bit WR to port address E000_00A4
16-bit WR to port address E000_00A6

RD access:

FIFO0: 32-bit RD from port adderss E000_00A0
or
16-bit RD from port address E000_00A0
16-bit RD from port address E000_00A2

FIFO1: 32-bit RD from port adderss E000_00A4
or
16-bit RD from port address E000_00A4
16-bit RD from port address E000_00A6

3.3.2.1 SgList Header Format

For outgoing packets, the local link module expects a 5 quadlet-header from cmdtxBuffer or sgList, and will process this 5-quadlets header, and send out a 4-quadlets header to Firewire 800 bus (complies with 1394 protocol).

For incoming packets, the header always has 4 quadlets.

Refer to Table 3-5 for descriptions of the fields used.

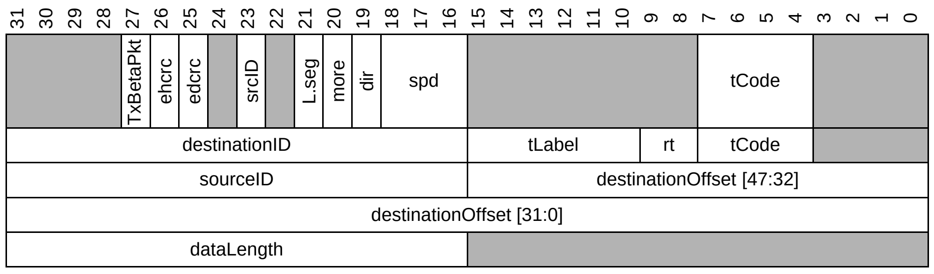


Figure 3-18 SgList Header Format

Table 3-5 SgList Header Format Field Definitions

Field Name	bit(s)	Description
1st Quadlet		
Reserved	31:28	RESERVED: These bits should be cleared to 0 by firmware during the initialization process.
TxBetaPkt	27	TRANSMIT BETA PACKET: This bit if set, the LINK_CORE sets the "RFMT" bit in the asynchronous packet transmit request pattern. Set this bit to explicitly request the PHY to use the Beta packet format for packet transmission.
ehcrc	26	HEADER CRC ERROR: If this bit is set to 1, the LINK_CORE sends the packet with a header CRC error.
edcrc	25	HEADER DATA ERROR: If this bit is set to 1, the LINK_CORE sends the packet with a data CRC error.
Reserved	24	RESERVED: This bit should be cleared to 0 by firmware during the initialization process.
srcID	23	SOURCE BUS ID SELECTOR: If cleared to 0, the high order 10 bits of the source_ID field of the transmitted packet will be 10’h3FF. If set to 1, the high order 10 bits of the source_ID field of the transmitted packet will be Node_ID.*busNumber*.
Reserved	22	RESERVED: This bit should be cleared to 0 by firmware during the initialization process.
L. Seg	21	LAST SEGMENT: If the command is the last one of S/G segments, this bit should be set to 1 by firmware.
more	20	MORE: If the number of commands in SgCmd buffer is more than one, this bit should be set to 1 by firmware.
dir	19	DIRECTION: When 0, the DMA data are transferred from the Firewire 800 bus to an ATA device. When 1, the DMA data are transferred from an ATA device to the Firewire 800 bus.
spd	18:16	SPEED: This field indicates the speed at which this packet is to be transmitted. 3’b000 = 100 Mbits/sec, 3’b001 = 200 Mbits/sec, 3’b010 = 400 Mbits/sec, and 3’b011 = 800 Mbits/sec. Other values are reserved.
tCode	7:4	TRANSACTION CODE: This is the transaction code for this packet.
2nd Quadlet		
destinationID	31:16	DESTINATION ID: This is the concatenation of the 10-bit bus number and the 6-bit node number for the Destination of this packet.
tLabel	15:10	TRANSACTION LABEL: This field is the transaction label, which is used to pair up a response packet with its corresponding request packet.
rt	9:8	RETRY CODE: This is the retry code for this packet. The LINK_CORE ignores the software provided retry code and substitutes an rt as appropriate for the implemented retry mechanism unless the rtsel bit is set to 1.
tCode	7:4	TRANSACTION CODE: This is the transaction code for this packet.
3rd & 4th Quadlets		
sourceID	31:16	SOURCE ID: This is the concatenation of the 10-bit bus number and the 6-bit node number for the Source of this packet.
destinationOffset [47:32] destinationOffset [31:0]	15:0 (2nd) 31:0 (3rd)	DESTINATION OFFSET: The concatenation of these two fields addresses a quadlet in the destination node’s Address space. This address must be quadlet-aligned (modulo 4).
5th Quadlet		
dataLength	31:16	DATA LENGTH: The number of bytes requested in a block read request or number of bytes in a block response packet.

3.4 Memories

Notes: To provide easy firmware programming and maximum flexibility, the decoding logic does not distinguish code space and data space. Therefore, **NO write protection** to code space is provided. The firmware programmer needs to be aware, and only write to firmware area when intended to do so.

In the tables below, **shadowEn** is reg. E000-00BAh, bit 7 and **shadowExt** is reg. E000-00BAh, bit 6.

3.4.1 External Flash Memory

CPU accesses Flash: using 8-bit or 16-bit modes. Refer to Section 3.1.1 for Firmware Shadow Procedure.

Address	Read Value	Write Value
0000_0000h- 01FF_FFFFh	(32 Mbytes) ~nRw && (**~shadowEn**) Initial Power On, start the code fetch from here	(32 Mbytes) nRw && (**~shadowEn**)
C000_0000h- C1FF_FFFFh	(32 Mbytes) ~nRw && (**shadowEn && ~shadowExt**) Partially shadowed by internal SRAM.	(32 Mbytes) nRw && (**shadowEn && ~shadowExt**)
4000_0000h- 41FF_FFFFh	(32 Mbytes) ~nRw && (**shadowEn && shadowExt**) Partially shadowed by external SRAM.	(32 Mbytes) nRw && (**shadowEn && shadowExt**)

3.4.2 Internal Memory Block

CPU accesses internal SRAM(16 Kbytes): C000_0000h - C000_3FFFh: WR/RD: use 8/16/32-bit mode.

Address	Read Value	Write Value
C000_0000h- C000_3FFFh	(16 Kbytes) ~nRW && ~(**shadowEn && ~shadowExt**)	(16 Kbytes) nRW && ~(**shadowEn && ~shadowExt**)
0000_0000h- 0000_3FFFh	(16K Bytes) ~nRW && (**shadowEn && ~shadowExt**) Mainly used as shadow RAM.	(16K Bytes) nRW && (**shadowEn && ~shadowExt**)

3.4.3 External Memory Block

CPU accesses external SRAM: WR/RD: using 16-bit mode**.**

Address	Read Value	Write Value
4000_0000h- 40FF_FFFFh	(16M Bytes) ~nRW && ~(**shadowEn && shadowExt**)	(16M Bytes) nRW && ~(**shadowEn && shadowExt**)
0000_0000h- 00FF_FFFFh	(16M Bytes) ~nRW && (**shadowEn && shadowExt**) Mainly used as shadow RAM.	(16M Bytes) nRW && (**shadowEn && shadowExt**)

4.1 Register Descriptions

NOTE: All registers have a *base address* of **0xE000-0000h**.
CPU accesses the internal registers using 8-bit or 16-bit or 32-bit modes.
CPU accesses internal buffers, SRAM using 32-bit mode.
CPU accesses Flash using 8-bit or 16-bit modes.

4.1.1 Link Block Registers

0x0000-0x0003 X Asynchronous Transmit Retries (XAT_Retry)

Bit(s)	rscu	reset	Acronym	Definition
31:29	rwu	3’h0	(secondLimit)	SECOND LIMIT:
28:16	rwu	13’h0	(cycleLimit)	CYCLE LIMIT:
				Together the secondLimit and cycleLimit fields define a time limit for retry attempts when the outbound dual-phase retry protocol is enabled. The secondLimit field represents a count in seconds modulo 8, and cycle-Limit represents a count in cycles modulo 8000. A value of zero written to both fields disables the outbound dual phase retry protocol and enables outbound single phase retry protocol.
15-4	r	12’h0	*reserved*	RESERVED
3:0	rw	4’h0	(maxATRetries)	MAX ASYNC TRANSMIT RETRIES: The RetryCountLimit field tells the 1394B_LINK Core how many times to attempt to retry the transmit operation for an asynchronous packet.

0x0004-0x0007 Y Asynchronous Transmit Retries (YAT_Retry)

Bit(s)	rscu	reset	Acronym	Definition
31:29	rwu	3’h0	(secondLimit)	SECOND LIMIT:
28:16	rwu	13’h0	(cycleLimit)	CYCLE LIMIT:
				Together the secondLimit and cycleLimit fields define a time limit for retry attempts when the outbound dual-phase retry protocol is enabled. The secondLimit field represents a count in seconds modulo 8, and cycle-Limit represents a count in cycles modulo 8000. A value of zero written to both fields disables the outbound dual phase retry protocol and enables outbound single phase retry protocol.
15-4	r	12’h0	*reserved*	RESERVED
3:0	rw	4’h0	(maxATRetries)	MAX ASYNC TRANSMIT RETRIES: The RetryCountLimit field tells the 1394B_LINK Core how many times to attempt to retry the transmit operation for an asynchronous packet.

0x0008-0x000B Z Asynchronous Transmit Retries (ZAT_Retry)

Bit(s)	rscu	reset	Acronym	Definition
31:29	rwu	3’h0	(secondLimit)	SECOND LIMIT:
28:16	rwu	13’h0	(cycleLimit)	CYCLE LIMIT: Together the secondLimit and cycleLimit fields define a time limit for retry attempts when the outbound dual-phase retry protocol is enabled. The secondLimit field represents a count in seconds modulo 8, and cycle-Limit represents a count in cycles modulo 8000. A value of zero written to both fields disables the outbound dual phase retry protocol and enables outbound single phase retry protocol.
15-4	r	12’h0	*reserved*	RESERVED
3:0	rw	4’h0	(maxATRetries)	MAX ASYNC TRANSMIT RETRIES: The RetryCountLimit field tells the 1394B_LINK Core how many times to attempt to retry the transmit operation for an asynchronous packet.

0x000C-0x000F TxAck Control (TxAckCntrl)

Bit(s)	rscu	reset	Acronym	Definition
31	rw	3’h0	(1’b0)	EXTERNAL ACKNOWLEDGE CODE ENABLE:
30-28	r	3’b0	*reserved*	RESERVED
27:24	rw	4’h0	(extAckCode)	EXTERNAL ACKNOWLEDGE CODE: When extAckCodeEn is set, the 1394B_LINK Core sends extAckCode[0:3] value as the acknowledge code for any non-broadcast asynchronous receive packet.
23-0	r	24’h0	*reserved*	RESERVED

0x0010-0x0013 Cycle Start Speed Control (CycStartSpdCtrl)

Bit(s)	rscu	reset	Acronym	Definition
31:28	rw	4’h0	(cycStrtTxSpd)	CYCLE START TRANSMIT SPEED: When the 1394B_LINK Core is the Cycle Master, it will transmit CSP according the speed value written into “cycStrtTxSpd” field. The mapping is as follows: 4’b0000 ⇐ S100 4’b0010 ⇐ S200 4’b0100 ⇐ S400 4’b0110 ⇐ S800
27:24	ru	4’h0	(cycStrtRxSpd)	CYCLE START RECEIVE SPEED: When the 1394B_LINK Core is the Cycle Slave, “cycStrtRxSpd” field will indicate the speed at which the 1394B_LINK Core received the last CSP.
23-0	r	24’h0	*reserved*	RESERVED

0x0018-0X001B Self ID Count (SelfIDCnt)

Bit(s)	rscu	reset	Acronym	Definition
31	ru	1’b0	(selfIDError)	SELF ID ERROR: This bit is set in the following situations: • “rxBufferFull” signal is asserted during Self-ID packet reception. • Self-ID packet reception starts before the 1394B_LINK Core writes all the quadlets of the synthesized bus reset packet into the Receive Interface. • A “selfIDErr” signal is asserted. • A parity error is detected in one of the receive Self-ID packets.
30-24	r	7’b0	*reserved*	RESERVED
23:16	ru	8’h00	(selfIDGeneration)	SELF ID GENERATION: The value in this field increments each time a receive Self-ID packet is written into the Receive Interface. This field rolls over to 0 after reaching 255.
15-13	r	3’b0	*reserved*	RESERVED
12:2	ru	11’h0	(selfIDSize)	SELF ID SIZE: This field indicates the number of quadlets present in the last receive Self-ID packet written by the 1394B_LINK Core to the Receive Interface. It includes the Self-ID header, Self-ID trailer and all the Self-ID packet quadlets received.
1-0	r	2’b0	*reserved*	RESERVED

0x001C-0x001F Reserved

0x0020-0x0023: set Channel Receive Hi (ChnlRcvHi_Set)
0x0024-0x0027: clr Channel Receive Hi (ChnlRcvHi_Clr)

Bit(s)	rscu	reset	Acronym	Definition
31:0	rsu	32’h0	(channelRcvHi)	CHANNEL RECEIVE HI: If channelRcvHi[n] is set, the 1394B_LINK Core will accept receive stream packets with channel number “n” and if channelRcvHi[n] is cleared, the 1394B_LINK Core will reject receive stream packets with channel number “n” where “n” can be 1, 2, 3, … and 31.

0x0028-0x002B: set Channel Receive Lo (ChnlRcvLo_Set)
0x002C-0x002F: clr Channel Receive Lo (ChnlRcvLo_Clr)

Bit(s)	rscu	reset	Acronym	Definition
31:0	rsu	32’h0	(channelRcvLo)	CHANNEL RECEIVE LO: If channelRcvLo[n] is set, the 1394B_LINK Core will accept receive stream packets with channel number “(n + 32)” and if channelRcvLo[n] is cleared, the 1394B_LINK Core will reject receive stream packets with channel number “(n + 32)” where “n” can be 1, 2, 3, … and 31.

0x0030-0x0033: set Interrupt Event (IntEvent_Set)
0x0034-0x0037: clr Interrupt Event (IntEvent_Clr)

This register reflects the state of the various interrupt sources from the 1394B_LINK Core. Reading the IntEvent_Set Register returns the current state of the IntEvent register. Reading the IntEvent_Clr Register returns the enabled version of the IntEvent register; i.e. the result of (IntEvent & IntEnable).

Bit(s)	rscu	reset	Acronym	Definition
31	rscu	1'b0	(asyXTxCompInt)	ASYXTXCOMP INTERRUPT: The asyXTxCompInt bit is set when the 1394B_LINK Core issues an asyXTxComp pulse.
30	rscu	1'b0	(asyYTxCompInt)	ASYYTXCOMP INTERRUPT: The asyYTxCompInt bit is set when the 1394B_LINK Core issues an asyYTxComp pulse.
29	rscu	1'b0	(asyZTxCompInt)	ASYZTXCOMP INTERRUPT: The asyZTxCompInt bit is set when the 1394B_LINK Core issues an asyZTxComp pulse.
28	rscu	1'b0	(*reserved*)	RESERVED
27	rscu	1'b0	(*reserved*)	RESERVED
26	rscu	1'b0	(asyXTxTCodeErrInt)	ASYXTXCODEERR INTERRUPT: The asyXTxCodrErrInt bit is set when the 1394B_LINK Core detects a reserved "tCode" value in the first quadlet of the packet in the asynchronous transmit queue X.
25	rscu	1'b0	(asyYTxTCodeErrInt)	ASYYTXCODEERR INTERRUPT: The asyYTxCodrErrInt bit is set when the 1394B_LINK Core detects a reserved "tCode" value in the first quadlet of the packet in the asynchronous transmit queue Y.
24	rscu	1'b0	(asyZTxTCodeErrInt)	ASYZTXCODEERR INTERRUPT: The asyZTxCodrErrInt bit is set when the 1394B_LINK Core detects a reserved "tCode" value in the first quadlet of the packet in the asynchronous transmit queue Z.
23	rscu	1'b0	(*reserved*)	RESERVED
22	rscu	1'b0	(*reserved*)	RESERVED
21	rscu	1'b0	(cycDataInvalidInt)	CYCLE DATA INVALID INTERRUPT: The cycleDataInvalidInt bit is set if the 1394B_LINK Core detects invalid Cycle Timer Contents in the received CSP (without header CRC error).
20	rscu	1'b0	(selfIDCompleteInt)	SELF ID COMPLETE INTERRUPT: The selfIDCompleteInt bit is set after the 1394B_LINK Core detects the first "subactionGap" event indication from PHY following a "busReset" event indication and after it has completely wrote the receive Self-ID packet to the Receive Interface.
19	rscu	1'b0	(ackMissing)	ACKNOWLEDGE MISSING: The ackMissing bit is set, after the transmission of a non-broadcast asynchronous primary packet if: • PHY indicates a "subactionGap" event before an ACK is received. • A parity error is detected in the ACK received. • A reserved acknowledge code is detected in the ACK received.
18	rscu	1'b0	(asyXRetryExcd)	ASYNCHRONOUS X RETRYS EXCEEDED: The asyXRetryExcd bit is set, if the last ACK received for the non-broadcast primary packet transmitted from asynchronous transmit queue X is ACK_BUSY_* and • if outbound single phase retry protocol is enabled, the number of retries attempted has reached the retry count limit set by AsyXATRetries.asyXRetryCountLimit field, or...

				• if outbound dual phase retry protocol is enabled, the retry time has reached the time limit set by AsyXATRetries.asyXSecondLimit and AsyXATRetries.asyXCycleLimit fields together.
17	rscu	1’b0	(asyYRetryExcd)	ASYNCHRONOUS Y RETRYS EXCEEDED: The asyYRetryExcd bit is set, if the last ACK received for the non-broadcast primary packet transmitted from asynchronous transmit queue Y is ACK_BUSY_* and • if outbound single phase retry protocol is enabled, the number of retries attempted has reached the retry count limit set by AsyYATRetries.asyYRetryCountLimit field, or... • if outbound dual phase retry protocol is enabled, the retry time has reached the time limit set by AsyYATRetries.asyYSecondLimit and AsyYATRetries.asyYCycleLimit fields together.
16	rscu	1’b0	(asyZRetryExcd)	ASYNCHRONOUS Z RETRYS EXCEEDED: The asyZRetryExceeded bit is set, if the last ACK received for the non-broadcast primary packet transmitted from asynchronous transmit queue Z is ACK_BUSY_* and • if outbound single phase retry protocol is enabled, the number of retries attempted has reached the retry count limit set by AsyZATRetries.asyZRetryCountLimit field, or... • if outbound dual phase retry protocol is enabled, the retry time has reached the time limit set by AsyZATRetries.asyZSecondLimit and AsyZATRetries.asyZCycleLimit fields together.
15	rscu	1’b0	(phyInterruptInt)	PHY_INTERRUPT INTERRUPT: The phyInterruptInt bit is set when the 1394B_LINK Core detects an Out-of-Band status transfer of type “PHY_INTERRUPT” over the pInt signal.
14	rscu	1’b0	(interfaceErrInt)	INTERFACE ERROR INTERRUPT: The interfaceErrorInt bit is set when the 1394B_LINK Core detects an Out-of-Band status transfer of type “INTERFACE_ERROR” over the pInt signal.
13	rscu	1’b0	(rxTCodeErrInt)	RX TCODE ERROR INTERRUPT: The rxTCodeErrInt bit is set when the 1394B_LINK Core detects a reserved “tCode” value in the first quadlet of the receive packet.
12	rscu	1’b0	(dataCrcErrInt)	DATA CRC ERROR INTERRUPT: The dataCrcErrInt is set when the 1394B_LINK Core detects a data CRC error in the receive packet.
11	rscu	1’b0	(hdrCRCErrInt)	HEADER CRC ERROR INTERRUPT: The hdrCrcErrInt is set when the 1394B_LINK Core detects a header CRC error in the receive packet.
10	rscu	1’b0	(phyRegRcvdInt)	PHY REGISTER RECEIVED INTERRUPT: The phyRegRcvdInt bit is set, when the 1394B_LINK Core detects an Out-of-Band status transfer of type “PHY_REGISTER_SOL” or “PHY_REGISTER_UNSOL” over the pInt signal.
9	rscu	1’b0	(phyResetInt)	PHY RESET INTERRUPT: The phyResetInt bit is set, when the 1394B_LINK Core detects an In-Band status transfer of type “PHY Interface Reset” over the dIn[0:7] signal lines.
8	rscu	1’b0	(busResetInt)	BUS RESET INTERRUPT: The busResetInt bit is set when the 1394B_LINK Core detects an In-Band status transfer of type “Bus Reset” over the dIn[0:7] signal lines.

7	rscu	1’b0	(restoreNoResetInt)	RESTORE_NO_RESET INTERRUPT: The restoreNoResetInt bit is set when the 1394B_LINK Core detects an Out-of-Band status transfer of type “PH_RESTORE_NO_RESET” over the pInt signal.
6	rscu	undef	(restoreResetInt)	RESTORE_RESET INTERRUPT: The restoreResetInt bit is set when the 1394B_LINK Core detects an Out-of-Band status transfer of type “PH_RESTORE_RESET” over the pInt signal.
5	rscu	undef	(cycleLostInt)	CYCLE LOST INTERRUPT: The cycleLostInt bit is set when the 1394B_LINK Core detects two consecutive “cycleSynch” events without receiving any CSP in between.
4	rscu	undef	(*reserved*)	RESERVED
3	rscu	1’b0	(subactionGapInt)	SUBACTION GAP INTERRUPT: The subactionGapInt bit is set when the 1394B_LINK Core detects an In-Band status transfer of type “Subaction Gap” over the dIn[0:7] signal lines.
2	rscu	1’b0	(arbResetGapInt)	ARBITRATION RESET GAP INTERRUPT: The arbResetGapInt bit is set when the 1394B_LINK Core detects an In-Band status transfer of type “Arbitration Reset Gap - Odd” or “Arbitration Reset Gap - Even” over the dIn[0:7] signal lines.
1	rscu	1’b0	(cycleTooLongInt)	CYCLE TOO LONG INTERRUPT: The cycleTooLongInt bit is set if the 1394B_LINK Core does not detect an In-Band status transfer of type “Subaction Gap” over the dIn[0:7] signal lines even after 118µs have elapsed after it last detected an In-Band status transfer of type “Cycle Start - Odd” or “Cycle Start - Even” over the dIn[0:7] signal lines. This indicates that the isochronous cycle lasted longer than 118µs.
0	rscu	1’b0	(commandRstInt)	COMMAND RESET INTERRUPT: The commandRstInt bit is set if the 1394B_LINK Core detects a valid Write Request for Data Quadlet packet addressed to the “RESET_START” register (destinationOffsetHigh = 16’hFFFF and destinationOffsetLow = 32’hF000_000C).

0x0038-0x003B: set Interrupt Enable (IntEnable_Set)
0x003C-0x003F: clr Interrupt Enable (IntEnable_Clr)

All the bits in the IntEnable register can be set and cleared. Only those bits, which are valid in the IntEvent register, are valid in IntEnable register also. For example, bit position 11 is reserved in both IntEvent and IntEnable registers. Any bit that is set in the IntEnable register will enable the interrupt event that occupies the same bit position in IntEvent register. For example, if bit [30] in IntEnable register is set, the interrupt event “cycleTooLongInt” (bit [30] in IntEvent register) will be enabled. Any bit that is cleared in the IntEnable register will disable the interrupt event that occupies the same bit position in IntEvent register. There are two addresses for this register: IntEnableSet and IntEnableClr. On read, both addresses return the contents of the IntEnable register.

0x0040-0x0043 Fairness Control (FairnessCntrl)

Bit(s)	rscu	reset	Acronym	Definition
31-6	r	26’h0	*reserved*	RESERVED
5:0	rw	6’h0	(priReqIn)	PRIORITY REQUEST IN: This field specifies the maximum number of “CUR_ASYNC” arbitration requests for asynchronous request packets that the 1394B_LINK Core is permitted to make of the PHY during a fairness interval. A priReqIn value of 6’h0 is equivalent to the behaviour specified by IEEE 1394-1995 Standard.

0x0044-0x0047 Ping Count (PingCntReg)

Bit(s)	rscu	reset	Acronym	Definition
31	rwu	1'B0	(pingCountValid)	PING COUNT VALID: When asserted, "pingCountValid" indicates that the contents of "pingCount" field are valid and were generated for the previous transmit PHY packet. The 1394B_LINK Core clears this bit when it starts transmission of a PHY packet.
30-10	r	26'h0	*reserved*	RESERVED
9:0	r	10'h0	(pingCount)	PING COUNT: This field specifies the number of 24.576 MHz clock cycles elapsed from the end of transmission of a PHY packet to either the start of reception of a new packet or detecting of "subactionGap" or a "busReset" indication by the the 1394B_LINK Core. If no response is received for a PHY packet before the "pingCount" reaches a value of 10'h3FF, this counter will not roll over to 10'h00, but will saturate. It is cleared by hardware when a next PHY packet is transmitted by link. This field is valid only if the "pingCountValid" signal is asserted.

0x0048-0x004B: set Link Control (LinkCtl_Set)
0x004C-0x004F: clr Link Control (LinkCtl_Clr)

The LinkControl register provides the control flags that enable and configure the link core protocol portions of the LINK_CORE. It contains controls for the receiver, and cycle timer. There are two addresses for this register: LinControlSet and LinkControlClr. When read, both addresses return the contents of the LinkControl register.

When write, the two addresses have different behavior: a '1' written to the LinkControlSet causes the corresponding bit in the LinkControl register to be set, while a '0' written to the LinkControlSet leaves the corresponding bit in the LinkControl register unaffected. On the other hand, a '1' written to the LinkControlClr causes the corresponding bit in the LinkControl register to be cleared, while a '0' written to the LinkControlClr leaves the corresponding bit in the LinkControl register unaffected.

Bit(s)	rscu	reset	Acronym	Definition
31	rsc	1'b0	(cycleMstrCapable)	CYCLE MASTER CAPABLE: If set, this bit indicates that the 1394B_LINK Core is capable of becoming a Cycle Master.
30	r	1'b0	(preRoot)	PREROOT: This bit indicates whether the 1394B_LINK Core was root or not before the last "busReset" event; set if it was the root and clear, if it was not the root.
29	rscu	1'b1	(cycleMaster)	CYCLE MASTER: When this bit is set, the 1394B_LINK Core will transmit a CSP on every cycleSynch event. When this bit is clear, the 1394B_LINK Core will receive CSP from FireWire 800 (1394) Serial Bus to maintain synchronisation with the node, which is sending them. This bit is automatically cleared when the IntEvent.cycleTooLongInt occurs and cannot be set until the IntEvent.cycleTooLongInt bit is cleared. This bit can be set only if NodeID.root is set. After a bus reset, this bit is automatically activated by hardware. After the Bus Reset,

```
if (NodeID.root == 1) {
   if (LinkControl.preRoot)
      {LinkControl.cycleMaster retain its prior value}
   else
      {LinkControl.cycleMaster = cycleMasterCapable}
}
else {LinkControl.cycleMaster = 0}
```

28	rsc	1’b0	(*reserved*)	RESERVED
27	rsc	1’b0	(rcvPhyPkt)	RECEIVE PHY PACKET ENABLE: When set, the 1394B_LINK Core will accept incoming PHY packets. This does not control either the receipt of Self-ID packets during the Self-ID Phase or the queuing of synthesised bus reset packets. This does control receipt of any Self-ID packets received outside the Self-ID Phase.
26	rsc	1’b0	(rcvSelfID)	RECEIVE SELF ID ENABLE: When set, the 1394B_LINK Core will accept the Self-ID packets received during Bus Initialisation phase.
25-8	r	18’b0	*reserved*	RESERVED
7	rsc	1’b1	(inboundRetryMode)	INBOUND RETRY MODE: When inboundRetryMode is set, inbound single-phase retry protocol is enabled. When cleared, inbound dual-phase retry mode is enabled.
6-2	r	5’b0	*reserved*	RESERVED
1	rsc	1’b0	(interfaceReset)	INTERFACE RESET: When interfaceReset is set, the 1394B_LINK Core will initiate a PHY-link Interface reset operation by driving LPS low as per P1394b Draft 1.2 specifications. This bit is set by the hardware if the “lpsReg” input signal is de-asserted for at least one clock period. This bit is cleared by the hardware when the PHY-link Interface Reset operation is over.
0	r	1’b0	*reserved*	RESERVED

0x0050-0x0053 Node ID (NodeID)

This register contains the FireWire 800 address for the node on which this chip resides. The 16-bit combination of busNumber and nodeNumber is referred to as the Node-ID. This register is updated autonomously by the hardware whenever the 1394B_LINK Core receives an Out-of-Band status transfer of type “PHY_REGISTER_UNSOL” or “PHY_REGISTER_SOL”, addressed to PHY Register 0.

Bit(s)	rscu	reset	Acronym	Definition
31	ru	1’b0	(IDValid)	ID VALID: This bit indicates whether the 1394B_LINK Core has a valid nodeNumber. It is cleared when a “busReset” event is detected and set again when the 1394B_LINK Core receives a valid nodeNumber (≠ 63) from the PHY through an Out-of-Band status transfer.
30	ru	1’b0	(root)	ROOT: This bit indicates whether the attached PHY is root. It is cleared when “busReset” event is detected and set again if bit 6 is set in the PHY register data received during the Out-of-Band status transfer.
29-28	r	2’b0	*reserved*	RESERVED
27	ru	1’b0	(CPS)	CABLE POWER STATUS: This bit indicates whether the PHY is reporting that cable power status is OK (VP 8V). It is cleared when a “busReset” event is detected and set again if bit 7 is set in the PHY register data received during the Out-of-Band status transfer.
26-14	r	2’b0	*reserved*	RESERVED
15:6	rwu	10’h3ff	(busNumber)	BUS NUMBER: This number is used to identify the specific FireWire 800 bus this node belongs to when multiple FireWire 800-compatible busses are connected via a bridge. The reset value of 10’h3ff is a specifically reserved value:

5:0	ru	6’h3f	(nodeNumber)	NODE NUMBER: This number is the physical node number established by the PHY during self-ID phase. It is automatically set to bits [0:5] in the PHY register data received during the Out-of-Band status transfer. The 1394B_LINK Core will not receive any packets if the nodeNumber is 63.

0x0054-0x0057 PHY Control (PhyCtl)

The PhyControl register is used to read or write a PHY register. To read a PHY register, the address of that PHY register is written to the regAddr field of PhyControl register and the rdPhyReg bit is set.

When the read request has been sent to the PHY (through the lReqOut pin), the rdPhyReg bit is cleared to ‘0’. When the PHY returns the register content (through a status transfer), the phyRegRdDone and then the IntEvent.phyReRcvd bit are set.

The address of the received PHY register is placed in the rdAddr fields.

The contents of the received PHY register is placed in the phyRegRdData fields.

Software shall serialise all PHY register read and write requests. Only after the current PHY register read or write operation completes, software can issue another PHY register read or write request. Also software shall not try to set rdPhyReg and wrPhyReg at the same time. If so, rdPhyReg will have priority and wrPhyReg will be ignored.

Bit(s)	rscu	reset	Acronym	Definition
31	rwu	1’b0	(rdDone)	READ DONE: This bit is set to ‘1’ by hardware when a PHY register transfer from PHY to LINK is done. This bit will be cleared to ‘0’ by software when either rdReg or wrReg is set to ‘1’.
30	rwu	1’b0	(unsolicited)	UNSOLICITED: The unsolicited bit is set whenever the 1394B_LINK Core receives an Out-of-Band status transfer of type “PHY_REGISTER_UNSOL” over the pInt signal.
29-28	r	2’h0	*reserved*	RESERVED
27:24	ru	4’h0	(rdAddr)	READ ADDRESS: This is the PHY Register Address received along with the last Out-of-Band status transfer of type “PHY_REGISTER_SOL” or “PHY_REGISTER_UNSOL” over the pInt signal.
23:16	ru	8’h0	(rdData)	READ DATA: This is the PHY Register Address received along with the last Out-of-Band status transfer of type “PHY_REGISTER_SOL” or “PHY_REGISTER_UNSOL” over the pInt signal.
15	rwu	1’b0	(rdReg)	READ PHY REGISTER: Set this bit to ‘1’ to initiate a read request to a PHY register. This bit is cleared when the 1394B_LINK Core receives an Out-of-Band status transfer of type “PHY_REGISTER_SOL” over pInt signal.
14	rwu	1’b0	(wrReg)	WRITE PHY REGISTER: Set this bit to ‘1’ to initiate a write request to a PHY register. This bit is cleared when the write request has been sent.
13-12	r	2’b0	*reserved*	RESERVED
11:8	rw	4’h0	(regAddr)	REGISTER ADDRESS: This is the address of the PHY register to be written to or read from.
7:0	rw	8’h0	(wrData)	WRITE DATA: This is the contents to be written to a PHY register. It is ignored when doing a read PHY register request.

0x0058-0x005F Reserved

4.1.2 ATA Block Registers

Note: The INIC-2430 CPU will access the ATA device registers at the addresses shown below. Check the device register specifications for the actual register and bit definitions.

0x0090 ATA Data (Data[15:0])

The data register is 16 bits wide.

Bit(s)	rscu	reset	Acronym	Definition
15:0	rw	16’h0	(Data[15:0])	DATA [15:0]: These are the ATA data bits [15:0].

0x0091 ATA Error/Features (Error/Features[15:0])

If this address is read by the host, the Error register is read. If this address is written by the host, the Features register is written.

Bit(s)	rscu	reset	Acronym	Definition
15-3	r	13’h0	*reserved*	RESERVED
2	rw	2’b0	(ABRT)	ABORT: When set, this bit indicates that the requested command has been command aborted.
1-0	r	2’b0	*reserved*	RESERVED

0x0092 ATA Sector Count (SectorCount)

The contents of this register are command dependent.

Bit(s)	rscu	reset	Acronym	Definition
15:0	rw	16’h0	(SectCnt[15:0)	SECTOR COUNT [15:0]: These are the ATA sector count bits [15:0].

0x0093 ATA Sector Number (SectorNumber)

The contents of this register are command dependent.

Bit(s)	rscu	reset	Acronym	Definition
15:0	rw	16’h0	(SectNum[15:0])	SECTOR NUMBER [15:0]: These are the ATA sector number bits [15:0].

0x0094 ATA Cylinder Low (CylinderLow)

The contents of this register are command dependent.

Bit(s)	rscu	reset	Acronym	Definition
15:0	rw	16’h0	(CylLow[15:0)	CYLINDER LOW [15:0]: These are the ATA cyclinder low bits [15:0].

0x0095 ATA Cylinder High (CylinderHigh)

The contents of this register are command dependent.

Bit(s)	rscu	reset	Acronym	Definition
15:0	rw	16’h0	(CylHigh[15:0)	CYLINDER HIGH [15:0]: These are the ATA cylinder high bits [15:0].

0x0096 ATA Device (Device)

Bit 4 selects the device. The other bits in this register are command dependent.

Bit(s)	rscu	reset	Acronym	Definition
15-5	r	11’h0	*reserved*	RESERVED
4	rw	1’b0	(DEV)	DEVICE: When cleared, this bit indicates that Device 0 has been selected and when set to ‘1’, Device 1 is selected.
3-0	r	4’b0	*reserved*	RESERVED

0x0097 ATA Status/Command (Status/Command)

If this address is read by the host, the Status register is read. If this address is written by the host, the Command register is written.

Bit(s)	rscu	reset	Acronym	Definition
15-8	r	8’h0	*reserved*	RESERVED
7	rw	1’b0	(BSY)	BUSY: When set, this bit indicates that the device is busy.
6	rw	1’b0	(DRDY)	DEVICE READY: When set, this bit indicates that the device is ready.
5-4	r	2’b0	*reserved*	RESERVED
3	rw	1’b0	(DRQ)	DATA REQUEST: When set, this bit indicates that the device is ready to transfer a word of data between the host and the device.
2-1	r	2’b0	*reserved*	RESERVED
0	rw	1’b0	(ERR)	ERROR: When set, this bit indicates that an error has occurred during execution of the previous command.

0x0098-0x009D Reserved

0x009E ATA AlternateStatus / DeviceControl (AltSts/DevCntrl)

If this address is read by the host, the Alternate Status register is read. If this address is written by the host, the Device Control register is written.

Bit(s)	rscu	reset	Acronym	Definition
15-8	r	8’h0	*reserved*	RESERVED
7	rw	1’b0	(HOB)	HIGH ORDER BYTE: This bit is defined by the 48-bit Address feature set. A write to any Command Block register shall clear the HOB bit to zero.
6-3	r	4’b0	*reserved*	RESERVED
2	rw	1’b0	(SRST)	HOST SOFTWARE RESET: When set, this bit is the host software reset bit.
1	r	1’b0	(nIEN)	INTERRUPT ENABLE: This bit is the enable bit for the device Assertion of INTRQ to the host.
0	rw	1’b0	(0)	Bit ‘0’ shall be cleared to zero.

0x009F Reserved

4.1.3 Data Port to WR/RD S/G FIFOs

0x00A0 FIFO 0 Data (FIFO0D[7:0])

Bit(s)	rscu	reset	Acronym	Definition
7:0	rw	8’h0	(Fifo0Data[7:0])	DMA FIFO 0 DATA [7:0]: Software can access DMA FIFO 0 through this register 0x0A3-0A0 (using 32-bit or 16-bit mode access).

0x00A1 FIFO 0 Data (FIFO0D[15:8])

Bit(s)	rscu	reset	Acronym	Definition
7:0	rw	8’h0	(Fifo0Data[15:8])	DMA FIFO 0 DATA [15:8]: Software can access DMA FIFO 0 through this register 0x0A3-0A0.

0x00A2 FIFO 0 Data (FIFO0D[23:16])

Bit(s)	rscu	reset	Acronym	Definition
7:0	rw	8’h0	(Fifo0Data[23:16])	DMA FIFO 0 DATA [23:16]: Software can access DMA FIFO 0 through this register 0x0A3-0A0 (using 16-bit mode access).

0x00A3 FIFO 0 Data (FIFO0D[31:24])

Bit(s)	rscu	reset	Acronym	Definition
7:0	rw	8’h0	(Fifo0Data[31:24])	DMA FIFO 0 DATA [31:24]: Software can access DMA FIFO 0 through this register 0x0A3-0A0.

0x00A4 FIFO 1 Data (FIFO1D[7:0])

Bit(s)	rscu	reset	Acronym	Definition
7:0	rw	8’h0	(Fifo1Data[7:0])	DMA FIFO 1 DATA [7:0]: Software can access DMA FIFO 0 through this register 0x0A3-0A0 (using 32-bit or 16-bit mode access).

0x00A5 FIFO 1 Data (FIFO1D[15:8])

Bit(s)	rscu	reset	Acronym	Definition
7:0	rw	8’h0	(Fifo1Data[15:8])	DMA FIFO 1 DATA [15:8]: Software can access DMA FIFO 0 through this register 0x0A3-0A0.

0x00A6 FIFO 1 Data (FIFO1D[23:16])

Bit(s)	rscu	reset	Acronym	Definition
7:0	rw	8’h0	(Fifo1Data[23:16])	DMA FIFO 1 DATA [23:16]: Software can access DMA FIFO 0 through this register 0x0A3-0A0 (using 16-bit mode access).

0x00A7 FIFO 1 Data (FIFO1D[31:24])

Bit(s)	rscu	reset	Acronym	Definition
7:0	rw	8’h0	(Fifo1Data[31:24])	DMA FIFO 1 DATA [31:24]: Software can access DMA FIFO 0 through this register 0x0A3-0A0.

4.1.4 Bridge General Registers

0x00B0 sgPioCmd Control (sgPCtrl)

Bit(s)	rscu	reset	Acronym	Definition
7-4	rw	1’b0	(PioX3-0Run)	PIOXnRUN [3:0]: These run bits are used in conjunction with sgRun register bits [3:0]. These bits are self cleared by hardware after being set. Before setting them, firmware needs to disable AtaDMAEn and enable pioXEn bit in ATA Enable Register (reg 0x00C4). Fill the package header into sgList, set the sgRun bit on sgRun register and the set the corresponding pioXRun bit. The hardware will start transmitting the FIFO data to/from ATA using pio transfer mechanism.
3-0	rw	1’b0	(Pio3-0Run)	PIOnRUN [3:0]: Start Transfer for PIO mode. When these bits are set, the hardware will start transmitting data from DMA FIFO to FireWire 800 or receiving data from FireWire 800 to DMA FIFO, based on the DIR bit setting. These bits are self-cleared by hardware after set. Before set this bit, the firmware needs to disable AtaDMAEn bit on DMA Control register, write the package header into the segment of sgCMD buffer, set RUN bit on sgCMD Control register, and fill data into FIFO data register.

0x00B1 FIFO Status (FifoSts)

Read: ‘1’ means the corresponding channel’s transaction is failed.

Writing a ‘1’ will clear the corresponding status bit.

Bit(s)	rscu	reset	Acronym	Definition
7	r	1’b0	(*reserved*)	RESERVED
6	r	1’b0	(Fifo1Full)	DMA FIFO 1 FULL: This bit is used to indicate the DMA FIFO 1 Full status.
5	r	1’b1	(Fifo1Empty)	DMA FIFO 1 EMPTY: This bit is used to indicate that the DMA FIFO 1 empty status.
4	rw	1’b0	(Fifo1Rst)	DMA FIFO 1 RESET: This bit is used to reset DMA FIFO 1. This bit is self-cleared by hardware after set.
3	r	1’b0	(*reserved*)	RESERVED
2	rw	1’b0	(Fifo0Full)	DMA FIFO 0 FULL: This bit is used to indicate the DMA FIFO 0 Full status.
1	r	1’b1	(Fifo0Empty)	DMA FIFO 0 EMPTY: This bit is used to indicate that the DMA FIFO 0 empty status.
0	rw	1’b0	(Fifo0Rst)	DMA FIFO 0 RESET: This bit is used to reset DMA FIFO 0. This bit is self-cleared by hardware after set.

0x00B2 GPIO Data (gpioData)

Bit(s)	rscu	reset	Acronym	Definition
7	rw	1’b1	(AtaRST1#)	ATA CHANNEL 1 RESET: This bit, when clear, will reset the ATA device.
6	rwu	1’b0	(AtaRST0#)	ATA CHANNEL 0 RESET: This bit, when clear, will reset the ATA device.
5:0	rwu	1’b0	(GPIOD[5:0])	GPIO DATA [5:0]: These bits are General Purpose I/O Data bits [5:0].

0x00B3 GPIO Control (gpioCtrl)

Bit(s)	rscu	reset	Acronym	Definition
7-6	r	2’b0	(*reserved*)	RESERVED
5:0	rw	6’hD	(GPIOCtrl[5:0])	GPIO CONTROL [5:0]: When set, the GPIO data is output.

0x00B4 Test Control 0 (TestCtrl0)

Bit(s)	rscu	reset	Acronym	Definition
7	rw	1’b0	(RstAtaReq)	ATA REQUEST RESET: When set, this bit will reset the current ATA request (not used in INIC-2430).
6	r	1’b0	(ALink)	LINK SELECT: 0: System uses BLink. 1: System uses ALink.
5	r	1’b1	*reserved*	RESERVED
4	rw	1’b0	(RS232En)	RS232 MODE ENABLE: When set, the RS232 test mode is enabled.
3	rw	1’b0	(TestEn)	TEST MODE ENABLE: When set, the device is in test mode.
2:0	rw	3’h0	(TestSel2-0)	TEST MODE SELECT: These 3 bits select specific internal signals to be routed to the device’s outputs during test mode.

0x00B5 Test Control 1 (TestCtrl1)

Bit(s)	rscu	reset	Acronym	Definition
7	r	1’b0	(mbistEn)	MEMORY BIST ENABLE: When set, Memory Built-In-Self-Test enabled.
6	r	1’b0	(mbistRst)	MEMORY BIST RESET: When set, Memory Built-In-Self-Test reset.
5-3	r	3’h000	*reserved*	RESERVED
2:0	rw	3’b000	(retryDly[2:0])	FIREWIRE 800 RETRY DELAY ENABLE [2:0]: When set, the hardware will wait for the selected time interval and then send the request packet out after receiving ack_busy. 000: no delay 001: 125 us (1394B), 250 us (1394A) 010: 250 us (1394B), 500 us (1394A) 011: 500 us (1394B), 1 ms (1394A) 100: 1 ms (1394B), 2 ms (1394A) 101: 2 ms (1394B), 4 ms (1394A) 110: 4 ms (1394B), 8 ms (1394A) 111: *reserved*

0x00B6 ATA I/O Cell Driving Control (DrvCtrl)

Bit(s)	rscu	reset	Acronym	Definition
7-5	r	3'h0	*reserved*	RESERVED
4	rw	1'b0	(GPIOSel)	ATA CH1 PIN SELECT: 0: use ATA CH1's pins for ATA CH1. 1: use ATA CH1's pins for GPIO[5:2].
3	rw	1'b0	(ataTSEn1)	ATA CH1 ENABLE: When set, ATA CH1's outputs are enabled.
2	rw	1'b0	(ataTSEn0)	ATA CH0 ENABLE: When set, ATA CH0's outputs are enabled.
1:0	rw	3'b000	(DrvSel[1:0])	ATA OUTPUT DRIVE SELECT [1:0]: When set, the ATA Output drives: 00: 4 mA 01: 6 mA 10: 8 mA 11: 10 mA

0x00B7 Agent 0 State (Agent0Stat)

Bit(s)	rscu	reset	Acronym	Definition
7	rw	1'b0	(HeartBeat_0)	HEARTBEAT 0: When this bit is set, it will generate a interrupt to indicate that chip received a WrReqQB packet for LUN0 HeartBeat. When firmware write '1', will clear it.
6	rw	1'b0	(Faststart_0)	FASTSTART 0: When this bit is set, it will generate a interrupt to indicate that chip received a WrReqDB packet for LUN0 Faststart. When firmware write '1', will clear it.
5	rw	1'b0	(USN_0)	UNSOLICITED STATUS ENABLE 0: When this bit is set, it will generate a interrupt to indicate that chip received a WrReqQB packet for LUN0 Unsolicited_Status_Enable. When firmware write '1', will clear it.
4	rw	1'b0	(Doorbell_0)	DOORBELL 0: When this bit is set, it will generate a interrupt to indicate that chip received a WrReqQB packet for LUN0 Doorbell. When firmware write '1', will clear it.
3	r	1'b0	*reserved*	RESERVED
2	rw	1'b0	(ORB_Pointer_0)	ORB POINTER 0: When this bit is set, it will generate a interrupt to indicate that chip received a WrReqDB packet for LUN0 ORB Pointer and two quadlets of ORB pointer will be re-route to cmdTx0 buffer. When firmware write '1', will clear it.
1	rw	1'b0	(Agent_Reset_0)	AGENT RESET 0: When this bit is set, it will generate a interrupt to indicate that chip received a WrReqQB packet for LUN0 Reset Fetch Agent. When firmware write '1', will clear it.
0	rw	1'b0	(Agent_State_0)	AGENT STATE 0: When this bit is set, it will generate a interrupt to indicate that chip received a WrReqQB packet for LUN0 Report Fetch Agent State. When firmware write '1', will clear it.

0x00B8 Agent 1 State (Agent1Stat)

Bit(s)	rscu	reset	Acronym	Definition
7	rw	1’b0	(HeartBeat_1)	HEARTBEAT 1: When this bit is set, it will generate a interrupt to indicate that chip received a WrReqQB packet for LUN1 HeartBeat. When firmware write ‘1’, will clear it.
6	rw	1’b0	(Faststart_1)	FASTSTART 1: When this bit is set, it will generate a interrupt to indicate that chip received a WrReqDB packet for LUN1 Faststart. When firmware write ‘1’, will clear it.
5	rw	1’b0	(USN_1)	UNSOLICITED STATUS ENABLE 1: When this bit is set, it will generate a interrupt to indicate that chip received a WrReqQB packet for LUN1 Unsolicited_Status_Enable. When firmware write ‘1’, will clear it.
4	rw	1’b0	(Doorbell_1)	DOORBELL 1: When this bit is set, it will generate a interrupt to indicate that chip received a WrReqQB packet for LUN1 Doorbell. When firmware write ‘1’, will clear it.
3	r	1’b0	*reserved*	RESERVED
2	rw	1’b0	(ORB_Pointer_1)	ORB POINTER 1: When this bit is set, it will generate a interrupt to indicate that chip received a WrReqDB packet for LUN1 ORB Pointer and two quadlets of ORB pointer will be re-route to cmdTx0 buffer. When firmware write ‘1’, will clear it.
1	rw	1’b0	(Agent_Reset_1)	AGENT RESET 1: When this bit is set, it will generate a interrupt to indicate that chip received a WrReqQB packet for LUN1 Reset Fetch Agent. When firmware write ‘1’, will clear it.
0	rw	1’b0	(Agent_State_1)	AGENT STATE 1: When this bit is set, it will generate a interrupt to indicate that chip received a WrReqQB packet for LUN1 Report Fetch Agent State. When firmware write ‘1’, will clear it.

0x00B9 Agent 1 Offset (Agent1Ofs)

Bit(s)	rscu	reset	Acronym	Definition
7:0	rw	8’h80	(AgentOfs[7:0])	FETCH AGENT OFFSET: Fetch Agent Offset for LUN1. This offset will be added to the base addesss of Fetch agent (FFFF F001 0000h) for pointering to LUN1 fetch agent registers.

0x00BA uP Control (uPCtrl)

Bit(s)	rscu	reset	Acronym	Definition
7	rw	1’b0	(ShadowEn)	SHADOW ENABLE: After power on, firmware will copy the code into shadow RAM. Then, firmware should set this bit to enable the shadow RAM to provide code to CPU.
6	rw	1’b0	(ShadowExt)	EXTERNAL SHADOW RAM: 0: Use internal SRAM as the shadow RAM (shadowEn needs to be 1). 1: Use external SRAM as the shadow RAM (shadowEn needs to be 1).
5-0	r	6’h0	*reserved*	RESERVED

0x00BB Power Management Control (PwrMgntCtrl)

The power management mechanism works as follows:

1 CPU writes 1 to register 0xBB bit 0 to shut down the clocks on most parts of the bridge. Clocks ON/OFF switching will occurs when clocks are at low voltage level.

2 CPU will enter its low power mode when its pipeline instruction queue is idle for 5 CLKs.

3 The system is now in "sleep".

4 4. An incoming "Phy resume packet" will wake up the bridge, restore the clocks, set the wakeup bit (register 0xBB bit 1), clear the sleep bit (register 0xBB bit 0), and optionally generate the sysINT to CPU (if register 0xBB bit 2 has been set to 1).

Bit(s)	rscu	reset	Acronym	Definition
7-2	r	6'h0	*reserved*	RESERVED
1	rw	1'b0	(wakeup)	WAKEUP: The incoming "Phy resume packet" will set this bit to 1, firmware may read this register to determine the cause of the sysINT. RD: wakeup status WR: write a '1' to force wakeup (resume clock)
0	rw	1'b0	(RegSleep)	REGISTER SLEEP MODE: Firmware sets this bit to 1 to stop the clock. Incoming "Phy resume packet" will clear this bit to 0 to restore the clocks. Firmware may also write a 0 to this bit to clear it.

0x00BC Revision ID (RevID)

Bit(s)	rscu	reset	Acronym	Definition
7:0	rw	8'h80	(RevID[7:0])	REVISION ID: This register is *read only*.

0x00C0 Link Control (LinkCtrl)

Bit(s)	rscu	reset	Acronym	Definition
7	rwu	1'b1	(DataByteSwap)	DATA BYTE SWAP: A soft reset and a bus reset shall not affect this bit. 0: Data quadlets are sent/received in big endian order. 1: Data quadlets are sent/received in little endian order.
6	rwu	1'b0	(CoreHardRst)	CORE HARD RESET: When 1, this bit will do the "Async Reset" to all linkCore's Flip-flops which use "coreClk" as the clock input.
5	rwu	1'b0	(PhyHardRst)	PHY HARD RESET: When 1, this bit will do the "Async Reset" to all linkCore's Flip-flops which use "PhyClk" as the clock input.
4	rwu	1'b0	(blockAckDataErr)	BLOCK ACK DATA ERROR: When 1, linkCore will send an ACK_BUSY_* instead of ACK_DATA_ERROR if a data CRC is detected for the incoming packet.
3	rwu	1'b1	LPS	LINK POWER STATUS CONTROL: This bit is used to control the Link Power Status. Software must set LPS to 1 to permit Link↔PHY communication. Once set, the link can use LREQs to perform PHY reads and writes. An LPS value of 0 prevents Link↔PHY communication. In this state, the only accessible Host Controller registers are Version, VendorID, HCControl, GUID_ROM, GUIDHi and GUIDLo. Access to other registers is not defined. Hardware and software resets clear LPS to 0. Software shall not clear LPS.

2	rwu	1'b0	(PostedWrEn)	POSTED WRITE ENABLE: When set to '1', physical posted writes. When '0', physical writes shall not be posted.
1	rwu	1'b0	(linkEnable)	LINK ENABLE: Software must set this bit to '1' when the system is ready to begin operation and then force a bus reset. This bit is necessary to keep other nodes from sending transactions before the local system is ready. When this bit is cleared the Host Controller is logically and immediately disconnected from the FireWire 800 bus. The link shall not process or interpret any packets received from the PHY, nor shall the link generate any bus requests. However, the link may access PHY registers via the PHY control register. This bit is cleared to '0' by hardware reset or software reset, and shall not be cleared by software. Software should not set the linkEnable bit until the Configuration ROM mapping register is valid.
0	rwu	1'b0	(softRst)	SOFT RESET: When set to '1', all Host Controller state is reset, all FIFO's are flushed, and Host Controller registers is reset. The read value of this bit is '1' while a soft reset or hard reset is in progress. The read value of this bit is '0' when neither soft reset nor hard reset is in progress. Software can use the value of his bit to determine when a reset has completed and the Host Controller is safe to operate.

0x00C1 DMA Control (DMACtrl)

Bit(s)	rscu	reset	Acronym	Definition
7-4	r	4'h0	*reserved*	RESERVED
3	rw	1'b1	(DIOW#)	DIOW: When DMA FIFO is underrun, this bit is used by firmware to toggle DIOW# signal.
2	rw	1'b1	(DIOR#)	DIOR: When DMA FIFO is underrun, this bit is used by firmware to toggle DIOR# signal.
1	rw	1'b1	(DMACK#)	DMACK: When DMA FIFO is underrun, this bit is used by firmware to toggle DMACK# signal.
0	rw	1'b0	(FlushAbort)	FLUSH / ABORT: When DMA FIFO is overrun, this bit is used by firmware to flush data out for outgoing data or abort the DMA operation for incoming data. This bit is self-cleared by hardware.

0x00C2-0x00C3 Reserved

0x00C4 DMA Control (DMACtrl)

Bit(s)	rscu	reset	Acronym	Definition
7	rw	1'b1	(AtaDMAEn)	ATA DMA ENABLE: This bit when set enables ataDMA.
6	rw	1'b0	(pioReqGrnt)	PIO REQ/GRANT: Write a 1 for PIO request. PIO grant status when read.
5	rw	1'b0	(pioXEn)	PIOX ENGINE ENABLE: This bit when set enables pioX engine. (AtaDMAEn should be disabled when pioX engine is enabled.)
4	rw	1'b0	(CFEn)	COMPACT FLASH ENABLE: When set to '1', this bit enables the Compact Flash.
3-0	r	4'h0	*reserved*	RESERVED

0x00C5 ATA Master Device Control (ataMstrCtrl)

Bit(s)	rscu	reset	Acronym	Definition
7:4	rw	4’h0	(dmaMode[3:0])	DMA MODE [3:0]: Note - uDMA mode 6 is only supported when running 1394b mode. 0000: DMA mode 0 (11 Mb/s) 0001: DMA mode 1 (13 Mb/s) 0010: DMA mode 2 (16 Mb/s) 1010: uDMA mode 2 (33 Mb/s) 1100: uDMA mode 4 (66 Mb/s) 1101: uDMA mode 5 (100 Mb/s) 1110: uDMA mode 6 (133 Mb/s) all other values: *reserved*
3:0	rw	4’h0	(pioMode[3:0])	PIO MODE [3:0]: 0000: PIO mode 0 0001: PIO mode 1 0010: PIO mode 2 0011: PIO mode 3 0100: PIO mode 4 all other values: *reserved*

0x00C6 ATA Slave Device Control (ataSlvCtrl)

Bit(s)	rscu	reset	Acronym	Definition
7:4	rw	4’h0	(dmaMode[3:0])	DMA MODE [3:0]: Note - uDMA mode 6 is only supported when running 1394b mode. 0000: DMA mode 0 (11 Mb/s) 0001: DMA mode 1 (13 Mb/s) 0010: DMA mode 2 (16 Mb/s) 1010: uDMA mode 2 (33 Mb/s) 1100: uDMA mode 4 (66 Mb/s) 1101: uDMA mode 5 (100 Mb/s) 1110: uDMA mode 6 (133 Mb/s) all other values: *reserved*
3:0	rw	4’h0	(pioMode[3:0])	PIO MODE [3:0]: 0000: PIO mode 0 0001: PIO mode 1 0010: PIO mode 2 0011: PIO mode 3 0100: PIO mode 4 all other values: *reserved*

0x00C7 ATA Control/Status (ataStatus)

Bit(s)	rscu	reset	Acronym	Definition
7	rw	1’b0	(ataCh1En)	ATA CHANNEL 1 ENABLE: When set, the ATA channel is switched to 1.
6	rw	1’b0	(ataCh01En)	ATA CHANNEL 0 &1 ENABLE: When set, DMA writes to channels 0 and 1 will be executed simultaneously.
5	ru	1’b0	(ataCh1Status)	ATA CHANNEL 1 STATUS: *Read Only*

4	ru	1’b0	(ataCh0Status)	ATA CHANNEL 0 STATUS: *Read Only*
3	ru	1’b0	(ataIORDY1)	ATA CHANNEL 1 IORDY: *Read Only*
2	ru	1’b0	(ataIORDY0)	ATA CHANNEL 0 IORDY: *Read Only*
1	ru	1’b0	(ataINTRQ1)	ATA CHANNEL 1 INTRQ: *Read Only*
0	ru	1’b0	(ataINTRQ0)	ATA CHANNEL 0 INTRQ: *Read Only*

0x00D0 Login ID0L (LoginID0L)

Bit(s)	**rscu**	**reset**	**Acronym**	**Definition**
7:0	rw	8’hff	(LoginID0[7:0])	LOGIN ID 0 [7:0]: These bits are the Login ID for LUN 0. This ID low byte is used by the hardware to recognize the incoming requests for LUN 0.

0x00D1 Login ID0H (LoginID0H)

Bit(s)	**rscu**	**reset**	**Acronym**	**Definition**
7:0	rw	8’hff	(LoginID0[15:8])	LOGIN ID 0 [15:8]: These bits are the Login ID for LUN 0. This ID high byte is used by the hardware to recognize the incoming requests for LUN 0.

0x00D2 Login ID1L (LoginID1L)

Bit(s)	**rscu**	**reset**	**Acronym**	**Definition**
7:0	rw	8’hff	(LoginID1[7:0])	LOGIN ID 1 [7:0]: These bits are the Login ID for LUN 1. This ID low byte is used by the hardware to recognize the incoming requests for LUN 1.

0x00D3 Login ID1H (LoginID1H)

Bit(s)	**rscu**	**reset**	**Acronym**	**Definition**
7:0	rw	8’hff	(LoginID1[15:8])	LOGIN ID 1 [15:8]: These bits are the Login ID for LUN 1. This ID high byte is used by the hardware to recognize the incoming requests for LUN 1.

0x00E0 ATA SG Write Threshold (ataWrSgThrshld)

Bit(s)	**rscu**	**reset**	**Acronym**	**Definition**
7:0	rw	8’h00	(ataWrSgThrshld[7:0])	ATA SG WRITE THRESHOLD [7:0]: This register defines the threshold value when ATA writes data into the S/G Buffer. When the WritePointer reaches this value, the S/G buffer starts transferring data to FireWire 800. Bit[7:0] corresponds to Adr[11:4], for example: a value of 0x01 means threshold is 16 bytes. A value of 0xff means 255 x 16 bytes. When the value is 0, it will disable this feature. For best performance, the faster node may have a small threshold. To avoid RdPointer outruns of the WritePointer, the slower node should have a larger threshold value. However, if an outrun occurs, the INIC-2430 will trigger a retry on the FireWire 800 bus.

0x00E1 Reserved

0x00E2 Faststart 0/1 Offset (Faststart0/1Ofs)

Bit(s)	rscu	reset	Acronym	Definition
7:4	rw	4'h2	(faststart1Ofs[3:0])	FASTSTART 1 OFFSET [3:0]: These bits correspond to Adr[11:8].
3:0	rw	4'h1	(faststart0Ofs[3:0])	FASTSTART 0 OFFSET [3:0]: These bits correspond to Adr[11:8].

0x00E3-0x00E4 Reserved

0x00E5 RS232 Baud Rate (baudRateSel)

Bit(s)	rscu	reset	Acronym	Definition
7:3	r	5'h0	*reserved*	RESERVED
2:0	rw	3'h0	(baudRateSel[2:0])	BAUD RATE SELECT [2:0]:

BAUD RATE SELECT [2:0]:	1394B_Mode	1394A_Mode
000:	9600	4800
001:	19200	9600
010:	38400	19200
011:	57600	---
100:	115200	---
101:	---	
110:	---	
111:	---	

0x00E6 RS232 Transfer Select (rs232XfrSel)

Bit(s)	rscu	reset	Acronym	Definition
7-6	r	2'h0	*reserved*	RESERVED
5:4	rw	2'h0	(startBitSel[1:0])	START BIT SELECT [1:0]: 00: 1 Start bit 01: 1 Start bit 10: 2 Start bits 11: 3 Start bits
3:2	rw	2'h0	(stopBitSel[1:0])	STOP BIT SELECT [1:0]: 00: 1 Stop bit 01: 1 Stop bit 10: 2 Stop bits 11: 3 Stop bits
1:0	rw	2'h0	(parityBitSel[1:0])	PARITY BIT SELECT [1:0]: 00: no Parity 01: odd Parity 10: even Parity 11: *reserved*

0x00E7 RS232 WrPort (rs232WrPort)

Bit(s)	rscu	reset	Acronym	Definition
7:0	rw	8’h00	(WrRS232Data[7:0])	RS232 WRITE PORT DATA [7:0]: For debugging purpose, the firmware will write debugging messages (in ASCI codes) to this port, and INIC-2430 will convert it to the serial data, and send it out through RS232 protocol.

0x00E8 RS232 WrPort Status (rs232WrCtrl)

Bit(s)	rscu	reset	Acronym	Definition
7	ru	1’b0	(RS232DataRdy)	RS232 DATA READY: After firmware write ASCI data to the RS232WrPort, the hardware will set this bit to indicate that the RS232WrPort (reg 0x00E7) is occupied, and firmware should wait until this bit is cleared by hardware (after RS232WrPort send out its data) before try to write next data to RS232WrPort. The firmware should check this bit, make sure it is 0 before write data to RS232WrPort.
6-0	r	7’h0	*reserved*	RESERVED

0x00E9 RS232 RdPort (rs232RdPort)

Bit(s)	rscu	reset	Acronym	Definition
7:0	r	8’h00	(RdRS232Data[7:0])	RS232 READ PORT DATA [7:0]: This register will latch the data which echos back from the other end’s RS232.

0x00EA RS232 RdPort Status (rs232RdCtrl)

Bit(s)	rscu	reset	Acronym	Definition
7	ru	1’b0	(RS232RdPortSts)	RS232RDPORT FULL/EMPTY STATUS: This bit will be set by hardware when incoming data is in RS232RdPort. This bit will be cleared by hardware after the firmware read the data out from RS232RdPort.
6-3	r	4’h0	*reserved*	RESERVED
2	ru	1’b0	(StartBitErr)	START BIT ERROR: This bit This bit indicates ths incoming data has a start-bit error.
1	ru	1’b0	(StopBitErr)	STOP BIT ERROR: This bit This bit indicates ths incoming data has a stop-bit error.
0	ru	1’b0	(ParityBitErr)	PARITY BIT ERROR: This bit This bit indicates ths incoming data has a parity-bit error.

0x00EB: Timer_count [7:0] (timerCnt[7:0])

Bit(s)	rscu	reset	Acronym	Definition
7:0	rw	8’h00	(timerCnt[7:0])	TIMER COUNT [7:0]: WR: Timer_count[7:0] RD: Will read out the “Current_timer_count[7:0]”, value after reset is 8’hff.

0x00EC: Timer_count [15:8] (timerCnt[15:8])

7:0	rw	8'h00	(timerCnt[15:8])	TIMER COUNT [15:8]: WR: Timer_count[15:8] RD: Will read out the "Current_timer_count[15:8]", value after reset is 8'hff.

The unit of timer count is 125us. For example: Timer_count[15:0] with a value of 0x000a means 1250us to timeout.

A value of Timer_count[15:0] = 0will disable the timer.
Timer_count[15:0] value after reset: 16'h0000
Current_timer_count[15:0] value after reset: 16'hffff

When timeout occurs, it will automatically set the Timeout bit (Register 0xED bit 0), and optionally generate the sysINT to the CPU (if Register 0xED bit 1 is set). Also, the timer will automatically reload the timer_count[15:0] and start the count down again.

0x00ED: Timer Timeout (timerTimeout)

Bit(s)	rscu	reset	Acronym	Definition
7	rw	1'b0	(Timer_sim)	TIMER SIMULATION: This bit when set enables the timer simulation using a much faster clock. (This bit is for internal use only.)
6	r	1'b0	*reserved*	RESERVED
5:4	rw	2'b0	(waitDly)	WAIT DELAY: Insert delay after an incoming Resp packet. 00: no delay 01: 5 us 10: 10 us 11: 20 us
3-1	r	3'h0	*reserved*	RESERVED
0	rwu	1'b0	(Timeout)	TIMEOUT: Timeout status when read (WR 1 to clear status).

0x00F0 Wait State CPU R/W Reg (Wait_state_ARM_r/w_Reg)

Bit(s)	rscu	reset	Acronym	Definition
7:4	rw	4'hf	(WS_CPU_rd_Reg)	Wait state for CPU to access internal registers/Buffers/SRAM (RD cycle). The value defines the number of wait states. For example: A value of 0x5 means 5 wait states.
3:0	rw	4'hf	(WS_CPU_wr_Reg)	Wait state for CPU to access internal registers/Buffers/SRAM (WR cycle). The value defines the number of wait states. For example: A value of 0x5 means 5 wait states.

0x00F1 Wait State CPU R/W External SRAM (Wait_state_ARM_r/w_SRAM_ext)

Bit(s)	rscu	reset	Acronym	Definition
7:4	rw	4'hf	(WS_CPU_rd_ext_SRAM)	Wait state for CPU to access external SRAM (RD cycle). The value defines the number of wait states. For example: A value of 0x5 means 5 wait states.

3:0	rw	4’hf	(WS_CPU_wr_ext_SRAM)	Wait state for CPU to access external SRAM (WR cycle). The value defines the number of wait states. For example: A value of 0x5 means 5 wait states.

0x00F2 Wait State CPU R/W Flash (Wait_state_ARM_r/w_Flash)

Bit(s)	rscu	reset	Acronym	Definition
7:4	rw	4’hf	(WS_CPU_rd_Flash)	Wait state for CPU to access external Flash (RD cycle). The value defines the number of wait states. For example: A value of 0x5 means 5 wait states.
3:0	rw	4’hf	(WS_CPU_wr_Flash)	Wait state for CPU to access external Flash (WR cycle). The value defines the number of wait states. For example: A value of 0x5 means 5 wait states.

0x00F3 Wait State CPU R/W Internal SRAM (Wait_state_ARM_r/w_SRAM_int)

Bit(s)	rscu	reset	Acronym	Definition
7:4	rw	4’hf	(WS_CPU_rd_int_SRAM)	Wait state for CPU to access internal SRAM (RD cycle). The value defines the number of wait states. For example: A value of 0x5 means 5 wait states.
3:0	rw	4’hf	(WS_CPU_wr_int_SRAM)	Wait state for CPU to access internal SRAM (WR cycle). The value defines the number of wait states. For example: A value of 0x5 means 5 wait states.

0x00F4 Wait State CPU R/W Internal Buffers (Wait_state_ARM_r/w_int_Buffers)

Bit(s)	rscu	reset	Acronym	Definition
7:4	rw	4’hf	(WS_CPU_rd_int_Buffers)	Wait state for CPU to access internal Buffers (RD cycle). The value defines the number of wait states. For example: A value of 0x5 means 5 wait states.
3:0	rw	4’hf	(WS_CPU_wr_int_Buffers)	Wait state for CPU to access internal Buffers (WR cycle). The value defines the number of wait states. For example: A value of 0x5 means 5 wait states.

0x00F5 Wait State CPU R/W Link (Wait_state_ARM_r/w_Link)

Bit(s)	rscu	reset	Acronym	Definition
7:4	rw	4’hf	(WS_CPU_rd_Link_Buffers)	Wait state for CPU to access Link registers (RD cycle). The value defines the number of wait states. For example: A value of 0x5 means 5 wait states.
3:0	rw	4’hf	(WS_CPU_wr_Link_Buffers)	Wait state for CPU to access Link registers (WR cycle). The value defines the number of wait states. For example: A value of 0x5 means 5 wait states (minimum value should be $\geq$ 2).

0x00F6-0x00FF Reserved

0x0100-0x010F Reserved

0x0110: set sg Run (sgRun_Set)
0x0111: clr sg Run (sgRun_Clr)

Bit(s)	rscu	reset	Acronym	Definition
7-5	r	3'h0	*reserved*	RESERVED
4	rw	1'b0	(wait_for_DMARQ)	WAIT FOR DMARQ: When set, there will be a wait until ataDMARQ asserted to send out RdReq packet.
3	rwu	1'b0	(sg3Run)	SG RUN BITS 3-0: When Set register is set by software, the correspond ing channel is ready to be transmitted. The hardware clears the run bit when the transfer of the corresponding channel is completed or completed with error. Software can set one of bit[3:0] on Clear register to clear the corresponding bit.
2	rwu	1'b0	(sg2Run)	
1	rwu	1'b0	(sg1Run)	
0	rwu	1'b0	(sg0Run)	

0x0112: clr sg Error/Clear (sgErr_Clr)

Bit(s)	rscu	reset	Acronym	Definition
7	w	1'b0	(sgChRst)	SG CHANNEL RESET: Writing a 1 to this bit will reset the DMA channel, and the state machine will be reset to idle.
6-4	r	3'h0	*reserved*	RESERVED
3	rwu	1'b0	(sg3Error)	SG ERROR BITS 3-0: When hardware detects an interrupt or an error, these bits will be set. Software should read the intEvent register to determine the problem and set one of bits [3:0] on the Clear register to clear the corresponding Error bit.
2	rwu	1'b0	(sg2Error)	
1	rwu	1'b0	(sg1Error)	
0	rwu	1'b0	(sg0Error)	

0x0113: clr sg Retry Exceeded/Clear (sgRtryExcd_Clr)

Bit(s)	rscu	reset	Acronym	Definition
7	rw	1'b0	(Infinite_retry)	INFININTE RETRY: Writing a 1 to this bit will enable the infinite_retry mode, which will prevent the generation of RetryExceeded.
6	rw	1'b0	(Resend_RdResp)	RESEND RDRESP: When 0, if reReqEvent occurs, hardware will re-generate RdReq packet to request the same packet again. When 1, if reReqEvent occurs, hardware assumes the Source Node will re-send the RdResp packet.
5-4	r	2'h0	*reserved*	RESERVED
3	rwu	1'b0	(sg3RtryExcd)	SG RETRY EXCEEDED BITS 3-0: Reading these bits returns the bit status. Writing a 1 to these bits clears the bits.
2	rwu	1'b0	(sg2RtryExcd)	
1	rwu	1'b0	(sg1RtryExcd)	
0	rwu	1'b0	(sg0RtryExcd)	

0x0114-0x0117 Reserved

0x0118: set Command Tx Run (CmdTxRun_Set)
0x0119: clr Command Tx Run (CmdTxRun_Clr)

Bit(s)	rscu	reset	Acronym	Definition
7-6	r	2’b0	*reserved*	RESERVED
5	rwu	1’b0	(CmdTx5Run)	COMMAND TRANSFER RUN BITS 5-0: The Set register is set by software and cleared by hardware when transfer of the corresponding channel is completed or completed with error. When the Clear register is set by software, the corresponding command transmit out channel is reset. CmdTx0 (LUN0) and CmdTx2 (LUN1) are used for requesting ORB. CmdTx1 (LUN0) and CmdTx3 (LUN1) are used for sending Status. CmdTx4, 5 are for general purposes.
4	rwu	1’b0	(CmdTx4Run)	
3	rwu	1’b0	(CmdTx3Run)	
2	rwu	1’b0	(CmdTx2Run)	
1	rwu	1’b0	(CmdTx1Run)	
0	rwu	1’b0	(CmdTx0Run)	

0x011A: clr Command Tx Error/Clear (CmdTxErr_Clr)

Bit(s)	rscu	reset	Acronym	Definition
7-6	r	2’b0	*reserved*	RESERVED
5	rwu	1’b0	(CmdTx5Err)	COMMAND TRANSFER ERROR BITS 5-0: When one of these bits are set by hardware, it indicates that the transaction of the corresponding channel is failed. When the bit is set by software, the corresponding error bit will be cleared.
4	rwu	1’b0	(CmdTx4Err)	
3	rwu	1’b0	(CmdTx3Err)	
2	rwu	1’b0	(CmdTx2Err)	
1	rwu	1’b0	(CmdTx1Err)	
0	rwu	1’b0	(CmdTx0Err)	

0x011B: clr Command Tx Retry Exceeded/Clear (CmdTxRtryExcd_Clr)

Bit(s)	rscu	reset	Acronym	Definition
7-6	r	2’b0	*reserved*	RESERVED
5	rwu	1’b0	(CmdTx5RtryExcd)	COMMAND TRANSFER RETRY EXCEEDED BITS 5-0: Reading these bits returns the bit status. Writing a 1 to these bits clears the bits.
4	rwu	1’b0	(CmdTx4RtryExcd)	
3	rwu	1’b0	(CmdTx3RtryExcd)	
2	rwu	1’b0	(CmdTx2RtryExcd)	
1	rwu	1’b0	(CmdTx1RtryExcd)	
0	rwu	1’b0	(CmdTx0RtryExcd)	

0x011C: Command Tx Channel Reset (CmdTxChRst)

Bit(s)	rscu	reset	Acronym	Definition
7-6	r	2’b0	*reserved*	RESERVED
5	rwu	1’b0	(CmdTx5ChRst)	COMMAND TRANSFER CHANNEL RESET BITS 5-0: Reading these bits returns CmdTxErr (same as read register 0x011A). Writing a 1 to these bits will reset the corresponding channel to the idle state.
4	rwu	1’b0	(CmdTx4ChRst)	
3	rwu	1’b0	(CmdTx3ChRst)	
2	rwu	1’b0	(CmdTx2ChRst)	
1	rwu	1’b0	(CmdTx1ChRst)	
0	rwu	1’b0	(CmdTx0ChRst)	

0x011D: clr Command Tx 2/0 Busy Status (CmdTx2/0_busy)

Bit(s)	rscu	reset	Acronym	Definition
7-2	r	’h0	*reserved*	RESERVED
1	rwu	1’b0	(CmdTx2_busy)	CMDTX2 BUSY: Reading this bit reads CmdTx2_busy status. Write a ‘1’ to this bit clear CmdTx2_busy.
0	rwu	1’b0	(CmdTx0_busy)	CMDTX0 BUSY: Reading this bit reads CmdTx0_busy status. Write a ‘1’ to this bit clear CmdTx0_busy.

Note: Any of the following conditions will set CmdTx2_busy to 1:

1. CmdTx2Run bit is set
   OR
2. Incoming WrReq packet is an ORB_PTR or Faststart to LUN1.

Any of the following conditions will set CmdTx0_busy to ‘1’:

1. CmdTx0Run bit is set
   OR
2. Incoming WrReq packet is an ORB_PTR or Faststart to LUN0.

Note: Once these bits are set, any further Faststart will be treated as a Doorbell (will set Doorbell (bit 4), not Faststart (bit 6), in registers 0x00B7h, 0x00B8h).

These busy bits can only be cleared by writing 1 to the corresponding bit of this register.

0x011E: clr Command Rx Active (CmdRxActive_Clr)

Bit(s)	rscu	reset	Acronym	Definition
7-3	r	5’h0	*reserved*	RESERVED
2	r	1’b0	(Rx_to_be_handled)	CMDRXBUFFER SELECT: This bit indicates which cmdRxBuffer has the older incoming packet, and should be handled by firmware first (this bit is *read only*). When ‘0’, the firmware should handle cmdRx0 first, and when ‘1’, the firmware should handle cmdRx1 first. This bit will be cleared when CmdRx1Active (bit 1) is cleared.
1 0	rwu rwu	1’b0 1’b0	(CmdRx1Active) (CmdRx0Active)	CMDRX1, 0 ACTIVE: When these bits are set by hardware, they indicate that data is available in the corresponding CmdRx buffer. When these bits are set by software, the corresponding Command Rx channel is reset. Write a ‘1’ to clear each of these bits.

0x011F Reserved

0x0120: Ack Retry Enable (AckRtry_En)

Bit(s)	rscu	reset	Acronym	Definition
7	rw	1’b0	(Ack_AdrErr_RtryEn)	ACK ADDRESS ERROR RETRY ENABLE: Write a 1 to enable an Ack_AdrErr to trigger a Retry.
6	rw	1’b1	(Ack_TypeErr_RtryEn)	ACK TYPE ERROR RETRY ENABLE: Write a 1 to enable an Ack_TypeErr to trigger a Retry.

5	rw	1’b1	(Ack_DataErr_RtryEn)	ACK DATA ERROR RETRY ENABLE: Write a 1 to enable an Ack_DataErr to trigger a Retry.
4	rw	1’b0	(Ack_CnflctErr_RtryEn)	ACK CONFLICT ERROR RETRY ENABLE: Write a 1 to enable an Ack_CnflctErr to trigger a Retry.
3	rw	1’b0	(Ack_Trdy_RtryEn)	ACK TRDY RETRY ENABLE: Write a 1 to enable an Ack_Trdy to trigger a Retry.
2	rw	1’b1	(Ack_Missing_RtryEn)	ACK MISSING RETRY ENABLE: Write a 1 to enable an Ack_Missing to trigger a Retry.
1-0	r	2’b0	*reserved*	RESERVED

0x0121: rCode Retry Enable (rCodeRtry_En)

Bit(s)	rscu	reset	Acronym	Definition
7	rw	1’b0	(rCode_AdrErr_RtryEn)	RCODE ADDRESS ERROR RETRY ENABLE: Write a 1 to enable an rCode_AdrErr to trigger a reReqEvent.
6	rw	1’b1	(rCode_TypeErr_RtryEn)	RCODE TYPE ERROR RETRY ENABLE: Write a 1 to enable an rCode_TypeErr to trigger a reReqEvent.
5	rw	1’b1	(rCode_DataErr_RtryEn)	RCODE DATA ERROR RETRY ENABLE: Write a 1 to enable an rCode_DataErr to trigger a reReqEvent.
4	rw	1’b0	(rCode_CnflctErr_RtryEn)	RCODE CONFLICT ERROR RETRY ENABLE: Write a 1 to enable an rCode_CnflctErr to trigger a reReqEvent.
3	rw	1’b1	(SplitTimeoutEn)	SPLIT TIMEOUT ENABLE: Write a 1 to enable a SplitTimeout to trigger a reReqEvent.
2	rw	1’b1	(DataCrcErr_RtryEn)	DATA CRC ERROR RETRY ENABLE: Write a 1 to enable a DataCrcErr to trigger a reReqEvent.
1	rw	1’b1	(HdrCrcErr_RtryEn)	HARDWARE CRC ERROR RETRY ENABLE: Write a 1 to enable an HdrCrcErr to trigger a reReqEvent.
0	r	1’b0	*reserved*	RESERVED: This bit is for manufacturing use only.

0x0122: reReq Channel Enable (reReq_Ch_En)

Bit(s)	rscu	reset	Acronym	Definition
7	rw	1’b1	(En_reReqSG)	SG CHANNEL RESEND ENABLE: Write a 1 to enable the SG channel to re-Send the previous Request packet if an reReqEvent occurs.
6	r	1’b1	*reserved*	RESERVED
5	rw	1’b1	(En_reReqTx5)	CMDTX5 CHANNEL RESEND ENABLE: Write a 1 to enable the cmdTx5 channel to re-Send the previous Request packet if an reReqEvent occurs.
4	rw	1’b1	(En_reReqTx4)	CMDTX4 CHANNEL RESEND ENABLE: Write a 1 to enable the cmdTx4 channel to re-Send the previous Request packet if an reReqEvent occurs.
3	rw	1’b1	(En_reReqTx3)	CMDTX3 CHANNEL RESEND ENABLE: Write a 1 to enable the cmdTx3 channel to re-Send the previous Request packet if an reReqEvent occurs.
2	rw	1’b1	(En_reReqTx2)	CMDTX2 CHANNEL RESEND ENABLE: Write a 1 to enable the cmdTx2 channel to re-Send the previous Request packet if an reReqEvent occurs.

1	rw	1’b1	(En_reReqTx1)	CMDTX1 CHANNEL RESEND ENABLE: Write a 1 to enable the cmdTx1 channel to re-Send the previous Request packet if an reReqEvent occurs.
0	rw	1’b1	(En_reReqTx0)	CMDTX0 CHANNEL RESEND ENABLE: Write a 1 to enable the cmdTx0 channel to re-Send the previous Request packet if an reReqEvent occurs.

0x0123-0x012F Reserved

0x0130: rCode / rCode (rCode/tCode)

Bit(s)	rscu	reset	Acronym	Definition
7-4	r	4’h0	(tCode[3:0])	TCODE [3:0]: These bits indicate the latest incoming packet’s tCode value.
3-0	r	4’h0	(rCode[3:0])	RCODE [3:0]: These bits indicate the latest incoming packet’s rCode value.

0x0131-0x013F Reserved

0x0140: Interrupt 0 Status (INT0_Status)

A read each of these bits returns the status of that bit. A write to a bit clears that bit.

Bit(s)	rscu	reset	Acronym	Definition
7	ruw	1’b0	(BusRst_INT)	BUS RESET INTERRUPT: BusRst_INT status from link.
6	rwu	1’b0	(RxDataRdy_INT)	RxDataRdy INTERRUPT: RxDataRdy_INT status from RS232 when incoming RS232 data is ready.
5	rwu	1’b0	(Wakeup_INT)	WAKEUP INTERRUPT: Wakeup_INT status from the power management logic.
4	rwu	1’b0	(Timeout_INT)	TIMEOUT INTERRUPT: Timeout_INT status from the Timer.
3	rwu	1’b0	(sg3RtryExcd_INT)	SG3 RETRY EXCEEDED INTERRUPT: sg3RtryExcd_INT status from sg3 channel.
2	rwu	1’b0	(sg2RtryExcd_INT)	SG2 RETRY EXCEEDED INTERRUPT: sg2RtryExcd_INT status from sg2 channel.
1	rwu	1’b0	(sg1RtryExcd_INT)	SG1 RETRY EXCEEDED INTERRUPT: sg1RtryExcd_INT status from sg1 channel.
0	rwu	1’b0	(sg0RtryExcd_INT)	SG0 RETRY EXCEEDED INTERRUPT: sg0RtryExcd_INT status from sg0 channel.

0x0141: Interrupt 1 Status (INT1_Status)

A read each of these bits returns the status of that bit. A write to a bit clears that bit.

Bit(s)	rscu	reset	Acronym	Definition
7	rwu	1’b0	(CmdRx1Actv_INT)	CMDRX1ACTV INTERRUPT: CmdRx1Actv_INT status from cmdRx1 channel.
6	rwu	1’b0	(CmdRx0Actv_INT)	CMDRX0ACTV INTERRUPT: CmdRx0Actv_INT status from cmdRx0 channel.

5	rwu	1’b0	(cmdTx5RtryExcd_INT)	CMDTX5 RETRY EXCEEDED INTERRUPT: cmdTx5RtryExcd_INT status from cmdTx5 channel.
4	rwu	1’b0	(cmdTx4RtryExcd_INT)	CMDTX4 RETRY EXCEEDED INTERRUPT: cmdTx4RtryExcd_INT status from cmdTx4 channel.
3	rwu	1’b0	(cmdTx3RtryExcd_INT)	CMDTX3 RETRY EXCEEDED INTERRUPT: cmdTx3RtryExcd_INT status from cmdTx3 channel.
2	rwu	1’b0	(cmdTx2RtryExcd_INT)	CMDTX2 RETRY EXCEEDED INTERRUPT: cmdTx2RtryExcd_INT status from cmdTx2 channel.
1	rwu	1’b0	(cmdTx1RtryExcd_INT)	CMDTX1 RETRY EXCEEDED INTERRUPT: cmdTx1RtryExcd_INT status from cmdTx1 channel.
0	rwu	1’b0	(cmdTx0RtryExcd_INT)	CMDTX0 RETRY EXCEEDED INTERRUPT: cmdTx0RtryExcd_INT status from cmdTx0 channel.

0x0142: Interrupt 2 Status (INT2_Status)

A read each of these bits returns the status of that bit. A write to a bit clears that bit.

Bit(s)	rscu	reset	Acronym	Definition
7	rwu	1’b0	(HeartBeat0_INT)	HEARTBEAT 0 INTERRUPT: HeartBeat0_INT status.
6	rwu	1’b0	(FastStart0_INT)	FASTSTART0 INTERRUPT: FastStart0_INT status.
5	rwu	1’b0	(USN0_INT)	USN0 INTERRUPT: USN0_INT status.
4	rwu	1’b0	(Doorbell0_INT)	DOORBELL0 INTERRUPT: Doorbell0_INT status.
3	rwu	1’b0	(GPIO0_INT)	GPIO0 INTERRUPT: GPIO0_INT status.
2	rwu	1’b0	(ORB_PTR0_INT)	ORB_PTR0 INTERRUPT: ORB_PTR0_INT status.
1	rwu	1’b0	(Agent0Rst_INT)	AGENT0 RESET INTERRUPT: Agent0Rst_INT status.
0	rwu	1’b0	(Agent0State_INT)	AGENT0 STATE INTERRUPT: Agent0State_INT status.

0x0143: Interrupt 3 Status (INT3_Status)

A read each of these bits returns the status of that bit. A write to a bit clears that bit.

Bit(s)	rscu	reset	Acronym	Definition
7	rwu	1’b0	(HeartBeat1_INT)	HEARTBEAT 1 INTERRUPT: HeartBeat1_INT status.
6	rwu	1’b0	(FastStart1_INT)	FASTSTART 1 INTERRUPT: FastStart1_INT status.
5	rwu	1’b0	(USN1_INT)	USN 1 INTERRUPT: USN1_INT status.
4	rwu	1’b0	(Doorbell1_INT)	DOORBELL 1 INTERRUPT: Doorbell1_INT status.
3	rwu	1’b0	(LinkOn_INT)	LINK ON INTERRUPT: LinkOn_INT status.
2	rwu	1’b0	(ORB_PTR1_INT)	ORB_PTR 1 INTERRUPT: ORB_PTR1_INT status.
1	rwu	1’b0	(Agent1Rst_INT)	AGENT1 RESET INTERRUPT: Agent1Rst_INT status.
0	rwu	1’b0	(Agent1State_INT)	AGENT1 STATE INTERRUPT: Agent1State_INT status.

0x0144-0x014F Reserved

0x0150: Interrupt 0 Enable (INT0_Enable)

Writing a 1 to a bit enables that bit's interrupt while writing a 0 to the bit masks that bit. A read of each of these bits returns the status of each bit.

Bit(s)	rscu	reset	Acronym	Definition
7	rw	1'b0	(BusRst_INT_En)	BUS RESET INTERRUPT ENABLE: BusRst_INT enable.
6	rw	1'b0	(RxDataRdy_INT_En)	RxDataRdy INTERRUPT ENABLE: RxDataRdy_INT enable.
5	rw	1'b0	(Wakeup_INT_En)	WAKEUP INTERRUPT ENABLE: Wakeup_INT enable.
4	rw	1'b0	(Timeout_INT_En)	TIMEOUT INTERRUPT ENABLE: Timeout_INT enable.
3	rw	1'b0	(sg3RtryExcd_INT_En)	SG3 RETRY EXCEEDED INTERRUPT ENABLE: sg3RtryExcd_INT enable.
2	rw	1'b0	(sg2RtryExcd_INT_En)	SG2 RETRY EXCEEDED INTERRUPT ENABLE: sg2RtryExcd_INT enable.
1	rw	1'b0	(sg1RtryExcd_INT_En)	SG1 RETRY EXCEEDED INTERRUPT ENABLE: sg1RtryExcd_INT enable.
0	rw	1'b0	(sg0RtryExcd_INT_En)	SG0 RETRY EXCEEDED INTERRUPT ENABLE: sg0RtryExcd_INT enable.

0x0151: Interrupt 1 Enable (INT1_Enable)

Writing a 1 to a bit enables that bit's interrupt while writing a 0 to the bit masks that bit. A read of each of these bits returns the status of each bit.

Bit(s)	rscu	reset	Acronym	Definition
7	rw	1'b0	(CmdRx1Actv_INT_En)	CMDRX1ACTV INTERRUPT ENABLE: CmdRx1Actv_INT enable.
6	rw	1'b0	(CmdRx0Actv_INT_En)	CMDRX0ACTV INTERRUPT ENABLE: CmdRx0Actv_INT enable.
5	rwu	1'b0	(cmdTx5RtryExcd_INT_En)	CMDTX5 RETRY EXCEEDED INTERRUPT ENABLE: cmdTx5RtryExcd_INT enable.
4	rw	1'b0	(cmdTx4RtryExcd_INT_En)	CMDTX4 RETRY EXCEEDED INTERRUPT ENABLE: cmdTx4RtryExcd_INT enable.
3	rw	1'b0	(cmdTx3RtryExcd_INT_En)	CMDTX3 RETRY EXCEEDED INTERRUPT ENABLE: cmdTx3RtryExcd_INT enable.
2	rw	1'b0	(cmdTx2RtryExcd_INT_En)	CMDTX2 RETRY EXCEEDED INTERRUPT ENABLE: cmdTx2RtryExcd_INT enable.
1	rw	1'b0	(cmdTx1RtryExcd_INT_En)	CMDTX1 RETRY EXCEEDED INTERRUPT ENABLE: cmdTx1RtryExcd_INT enable.
0	rw	1'b0	(cmdTx0RtryExcd_INT_En)	CMDTX0 RETRY EXCEEDED INTERRUPT ENABLE: cmdTx0RtryExcd_INT enable.

0x0152: Interrupt 2 Enable (INT2_Enable)

Writing a 1 to a bit enables that bit's interrupt while writing a 0 to the bit masks that bit. A read of each of these bits returns the status of each bit.

Bit(s)	rscu	reset	Acronym	Definition
7	rw	1'b0	(HeartBeat0_INT_En)	HEARTBEAT 0 INTERRUPT ENABLE: HeartBeat0_INT enable.
6	rw	1'b0	(FastStart0_INT_En)	FASTSTART0 INTERRUPT ENABLE: FastStart0_INT enable.
5	rw	1'b0	(USN0_INT_En)	USN0 INTERRUPT ENABLE: USN0_INT enable.
4	rw	1'b0	(Doorbell0_INT_En)	DOORBELL0 INTERRUPT ENABLE: Doorbell0_INT enable.
3	rw	1'b0	(GPIO0_INT_En)	GPIO0 INTERRUPT ENABLE: GPIO0_INT enable.
2	rw	1'b0	(ORB_PTR0_INT_En)	ORB_PTR0 INTERRUPT ENABLE: ORB_PTR0_INT enable.
1	rw	1'b0	(Agent0Rst_INT_En)	AGENT0 RESET INTERRUPT ENABLE: Agent0Rst_INT enable.
0	rw	1'b0	(Agent0State_INT_En)	AGENT0 STATE INTERRUPT ENABLE: Agent0State_INT enable.

0x0153: Interrupt 3 Enable (INT3_Enable)

Writing a 1 to a bit enables that bit's interrupt while writing a 0 to the bit masks that bit. A read of each of these bits returns the status of each bit.

Bit(s)	rscu	reset	Acronym	Definition
7	rw	1'b0	(HeartBeat1_INT_En)	HEARTBEAT 1 INTERRUPT ENABLE: HeartBeat1_INT enable.
6	rw	1'b0	(FastStart1_INT_En)	FASTSTART 1 INTERRUPT ENABLE: FastStart1_INT enable.
5	rw	1'b0	(USN1_INT_En)	USN 1 INTERRUPT ENABLE: USN1_INT enable.
4	rw	1'b0	(Doorbell1_INT_En)	DOORBELL 1 INTERRUPT ENABLE: Doorbell1_INT enable.
3	rw	1'b0	(LinkOn_INT_En)	LINK ON INTERRUPT ENABLE: LinkOn_INT enable.
2	rw	1'b0	(ORB_PTR1_INT_En)	ORB_PTR 1 INTERRUPT ENABLE: ORB_PTR1_INT enable.
1	rw	1'b0	(Agent1Rst_INT_En)	AGENT1 RESET INTERRUPT ENABLE: Agent1Rst_INT enable.
0	rw	1'b0	(Agent1State_INT_En)	AGENT1 STATE INTERRUPT ENABLE: Agent1State_INT enable.

0x0154: Trailer 0 Byte 0 (Trailer0_Byte0)

The bits in this register are ***read only***.

Bit(s)	rscu	reset	Acronym	Definition
7-0	r	8'h0	(Time0Stamp00)	TRAILER 0 QUADLET BYTE 0: Read the incoming packet's Trailer 0 quadlet's byte 0. The incoming packet itself is put into cmdRx0Buffer.

0x0155: Trailer 0 Byte 1 (Trailer0_Byte1)

The bits in this register are ***read only***.

Bit(s)	rscu	reset	Acronym	Definition
7-0	r	8'h0	(Time0Stamp01)	TRAILER 0 QUADLET BYTE 1: Read the incoming packet's Trailer 0 quadlet's byte 1. The incoming packet itself is put into cmdRx0Buffer.

0x0156: Trailer 0 Byte 2 (Trailer0_Byte2)

The bits in this register are ***read only***.

Bit(s)	rscu	reset	Acronym	Definition
7-0	r	8’h0	(Time0Stamp10)	TRAILER 0 QUADLET BYTE 2: Read the incoming packet’s Trailer 0 quadlet’s byte 2. The incoming packet itself is put into cmdRx0Buffer.

0x0157: Trailer 0 Byte 3 (Trailer0_Byte3)

The bits in this register are ***read only***.

Bit(s)	rscu	reset	Acronym	Definition
7-0	r	8’h0	(Time0Stamp11)	TRAILER 0 QUADLET BYTE 3: Read the incoming packet’s Trailer 0 quadlet’s byte 3. The incoming packet itself is put into cmdRx0Buffer.

0x0158: Trailer 1 Byte 0 (Trailer1_Byte0)

The bits in this register are ***read only***.

Bit(s)	rscu	reset	Acronym	Definition
7-0	r	8’h0	(Time1Stamp00)	TRAILER 1 QUADLET BYTE 0: Read the incoming packet’s Trailer 1 quadlet’s byte 0. The incoming packet itself is put into cmdRx1Buffer.

0x0159: Trailer 1 Byte 1 (Trailer1_Byte1)

The bits in this register are ***read only***.

Bit(s)	rscu	reset	Acronym	Definition
7-0	r	8’h0	(Time1Stamp01)	TRAILER 1 QUADLET BYTE 1: Read the incoming packet’s Trailer 1 quadlet’s byte 1. The incoming packet itself is put into cmdRx1Buffer.

0x015A: Trailer 1 Byte 2 (Trailer1_Byte2)

The bits in this register are ***read only***.

Bit(s)	rscu	reset	Acronym	Definition
7-0	r	8’h0	(Time1Stamp10)	TRAILER 1 QUADLET BYTE 2: Read the incoming packet’s Trailer 1 quadlet’s byte 2. The incoming packet itself is put into cmdRx1Buffer.

0x015B: Trailer 1 Byte 3 (Trailer1_Byte3)

The bits in this register are ***read only***.

Bit(s)	rscu	reset	Acronym	Definition
7-0	r	8’h0	(Time1Stamp11)	TRAILER 1 QUADLET BYTE 3: Read the incoming packet’s Trailer 1 quadlet’s byte 3. The incoming packet itself is put into cmdRx1Buffer.

0x015C: Miscellanous Enable (MiscEn)

The bits in this register are ***reserved***.

Bit(s)	rscu	reset	Acronym	Definition
7-0	rw	8’h03	*reserved*	RESERVED: These bits are used for internal testing purposes and are reserved.

0x015D-0x015F Reserved

0x0160-0x016F Reserved

0x0170: Split Timer Lo (Split_Timer_L)

Bit(s)	rscu	reset	Acronym	Definition
7-0	w	8’h0	(Split_Timer_L)	SPLIT TIMER [7:0]: Bits 7-0 of Split_Timer[15:0]. Split_Timer[15:0] specifies the Timeout value. If the Request packet has received Ack_pending and waits for more than the time duration specified by Split_Timer[15:0] without receiving its response packet, a Split_Timeout event will occur, which will trigger the hardware to re-send the Request packet. The resolution is 125us, a value of Split_Timer[15:0] = 16’h0005 means 5 x 125us = 575us. A value of 16’h0000 will disable this timer, and will not generate any Split_Timer event.

0x0171: Split Timer Hi (Split_Timer_H)

Bit(s)	rscu	reset	Acronym	Definition
7-0	w	8’h0	(Split_Timer_H)	SPLIT TIMER [15:8]: Bits 15-8 of Split_Timer[15:0]. Split_Timer[15:0] specifies the Timeout value. If the Request packet has received Ack_pending and waits for more than the time duration specified by Split_Timer[15:0] without receiving its response packet, a Split_Timeout event will occur, which will trigger the hardware to re-send the Request packet. The resolution is 125us, a value of Split_Timer[15:0] = 16’h0005 means 5 x 125us = 575us. A value of 16’h0000 will disable this timer, and will not generate any Split_Timer event.

SECTION 5
Electrical Specifications

5.1 Absolute Maximum Ratings

Table 5-1 Absolute Maximum Ratings

Parameter		Minimum	Maximum	Units
Environment	Storage Temperature	-40	150	°C
	Operating Temperature	0	115	°C
	ESD Immunity	2.0 KV human model		
Voltage Levels	I/O Logic Power Supply (3.3V)	-0.3	3.9	V
	Core Logic Power Supply (2.5V)	-0.3	3.0	V
	Inputs	-0.3	Vcc+0.3	V
	Outputs	-0.3	Vcc+0.3	V

NOTE: Stresses above those listed under “Absolute Maximum Ratings” may cause permanent damage to the device. This is a stress rating only and the functional operation of the device at these or any other conditions above those indicated in the operational sections of the specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

5.2 Recommended Operating Conditions

Ta = 0°C to +115°C

VCC_3.3 = 3.3V ± 5%

VCC_2.5 = 2.5V ± 5%

GND = 0V

Table 5-2 Device Recommended Operating Conditions

Symbol	Parameter	Minimum	Typical	Maximum	Units
VCC_3.3	I/O Logic Supply Voltage	3.0	3.3	3.6	V
VCC_2.5	Core Logic Supply Voltage	2.25	2.5	2.75	V
T_J	Commercial Junction Operating Temperature	0	25	115	°C
T_J	Industrial Junction Operating Temperature	-40	25	125	°C

5.3 General DC Characteristics

Ta = 0°C to +115°C

VCC_3.3 = 3.3V ± 5%

VCC_2.5 = 2.5V ± 5%

GND = 0V

Table 5-3 General DC Characteristics

Symbol	Parameter	Minimum	Typical	Maximum	Units
I_{IL}	Input Leakage Current	-10	-	10	µA
I_{OZ}	Tri-state Leakage Current	-10	-	10	µA
C_{IN}	Input Capacitance		3.1		pF
C_{OUT}	Output Capacitance	2.7	3.1	4.9	pF
C_{BID}	Bi-directional Buffer Capacitance	2.7	3.1	4.9	pF

5.4 DC Electrical Characteristics for Normal Operation

Ta = 0°C to +115°C

VCC = 3.3V ± 5%

VCC_2.5 = 2.5V ± 5%

GND = 0V

Table 5-4 DC Electrical Characteristics for Normal Operation

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Units
V_{IL}	Input Low Voltage	CMOS			0.3*Vcc	V
V_{IH}	Input High Voltage	CMOS	0.7*Vcc			V
V_{OL}	Output Low Voltage	I_{OH} = 2-24 mA			0.4	V
V_{OH}	Output High Voltage	I_{OH} = 2-24 mA	2.4			V
Ri	Input Pull Up/Down Resistance	V_{IL} = 0 / V_{IH} = Vcc	40	75	190	kΩ
I_{CC}	Operating Supply Current	VCC = 3.3V VCC_2.5 = 2.5V		30 50		mA

SECTION 6
Timing Specifications

6.1 1394 Link to PHY Interface

Symbol	Parameter	Minimum	Maximum	Units
T_{PSU}	CTL and D setup time to phy PCLK	2	-	ns
T_{PH}	CTL and D hold time after phy PCLK	2	-	ns
T_{LSU}	Delay time of driving CTL and D from phy LCLK	1.5	2.9	ns
T_{LH}	Delay time of driving CTL and D after phy LCLK	1.5	2.9	ns
T_{LHZ}	Delay time of driving CTL and D to high Z	1.5	2.9	ns

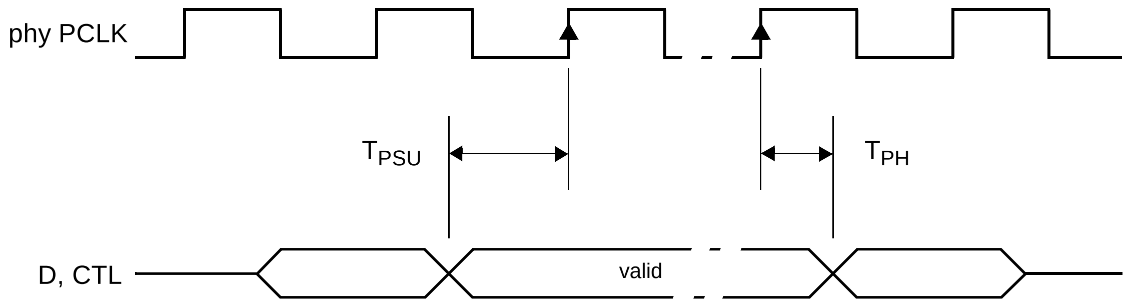


Figure 6-1 PHY to LINK Interface

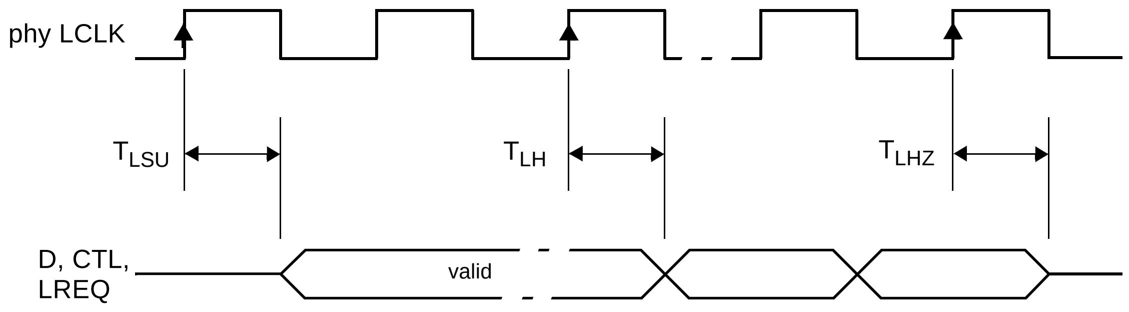


Figure 6-2 LINK to PHY Interface

6.2 Flash / Memory Interface

6.2.1 Flash Memory Read Cycle

Symbol	Parameter	Minimum	Maximum	Units
T_{PA}	phyPCLK to uA address valid time	7.0	9.0	ns
T_{PF}	phyPCLK to FlashRd active time	5.5	8.0	ns
T_{PN}	phyPCLK to FlashRd negated time	5.5	8.0	ns
T_{PD}	phyPCLK to uD read data valid time	see Flash memory spec		ns
T_{FD}	uD read data hold time from FlashRd negated	see Flash memory spec		ns

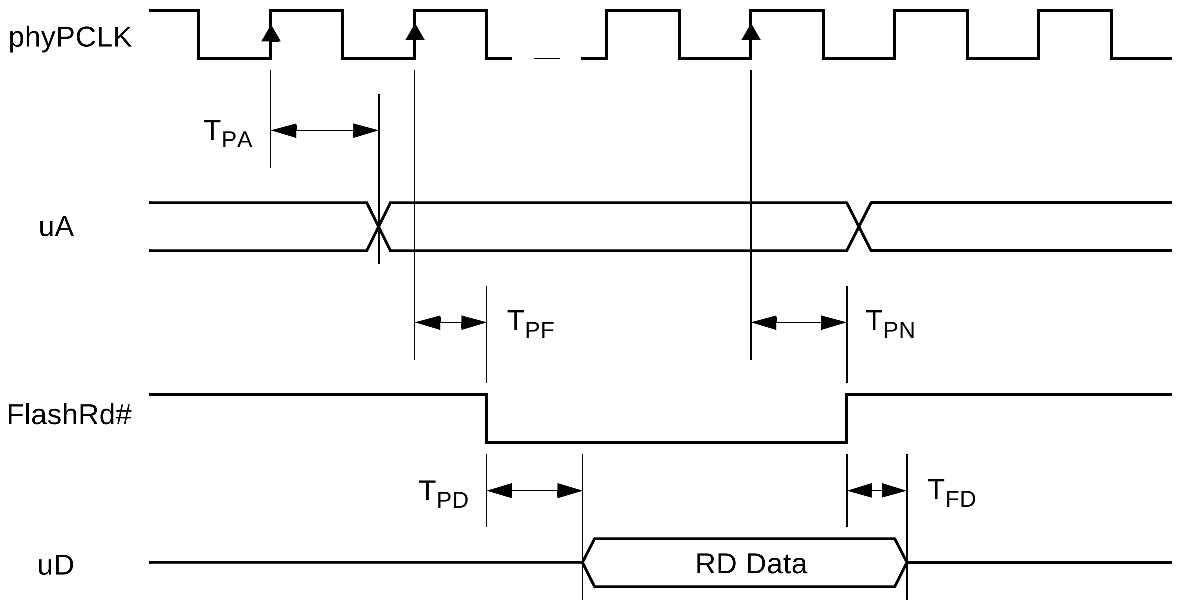


Figure 6-3 Flash Memory Read Cycle

6.2.2 Flash Memory Write Cycle

Symbol	Parameter	Minimum	Maximum	Units
T_{PA}	phyPCLK to uA address valid time	7.0	9.0	ns
T_{PF}	phyPCLK to FlashWr active time	5.5	8.0	ns
T_{PN}	phyPCLK to FlashWr negated time	5.5	8.0	ns
T_{PW}	phyPCLK to uD write data valid time	0	1.0	ns
T_{PWH}	phyPCLK to uD write data hold time	6.0	9.0	ns

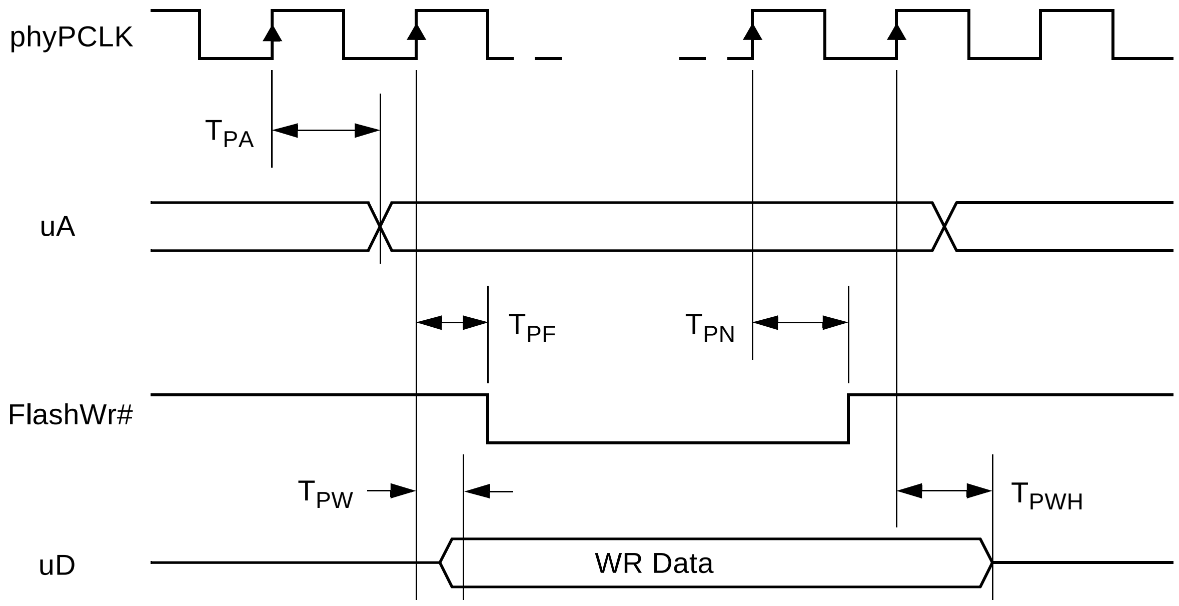


Figure 6-4 Flash Memory Write Cycle

6.2.3 External SRAM Read Cycle

Symbol	Parameter	Minimum	Maximum	Units
T_{PA}	phyPCLK to uA address valid time	7.0	9.0	ns
T_{PS}	phyPCLK to extSRAMRd active time	4.9	7.4	ns
T_{PN}	phyPCLK to extSRAMRd negated time	5.0	7.5	ns
T_{PD}	phyPCLK to uD read data valid time	see SRAM memory spec		ns
T_{DS}	uD read data hold time from extSRAMRd negated	see SRAM memory spec		ns

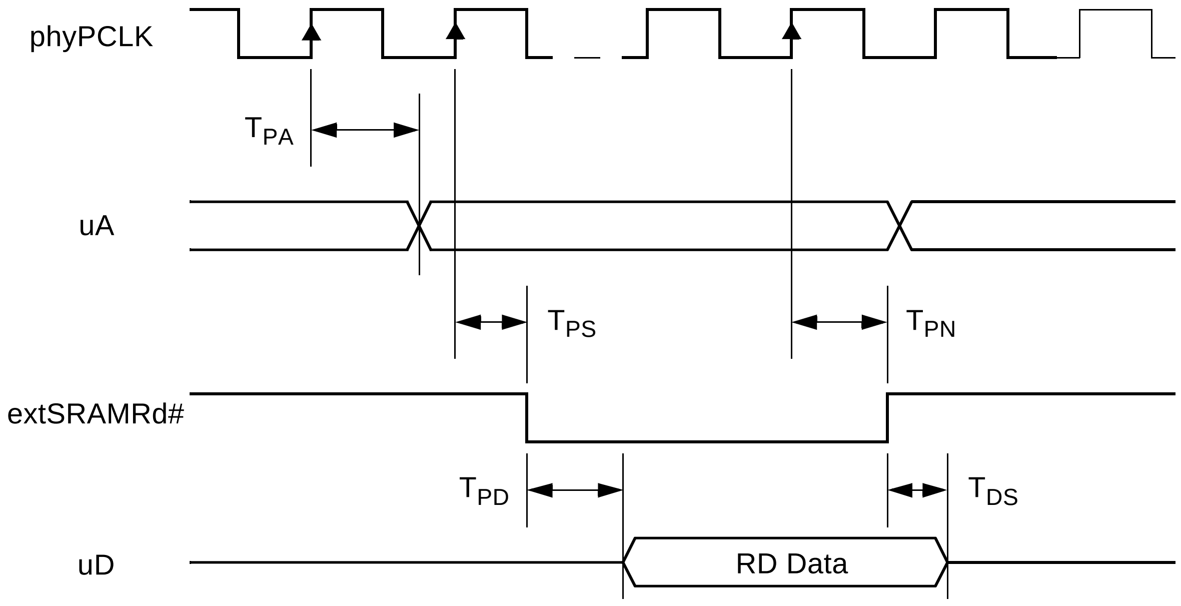


Figure 6-5 External SRAM Read Cycle

6.2.4 External SRAM Write Cycle

Symbol	Parameter	Minimum	Maximum	Units
T_{PA}	phyPCLK to uA address valid time	7.0	9.0	ns
T_{PS}	phyPCLK to extSRAMWr active time	5.4	8.1	ns
T_{PN}	phyPCLK to extSRAMWr negated time	5.5	8.3	ns
T_{PW}	phyPCLK to uD write data valid time	1.6	2.4	ns
T_{PWH}	phyPCLK to uD write data hold time	6.0	9.0	ns

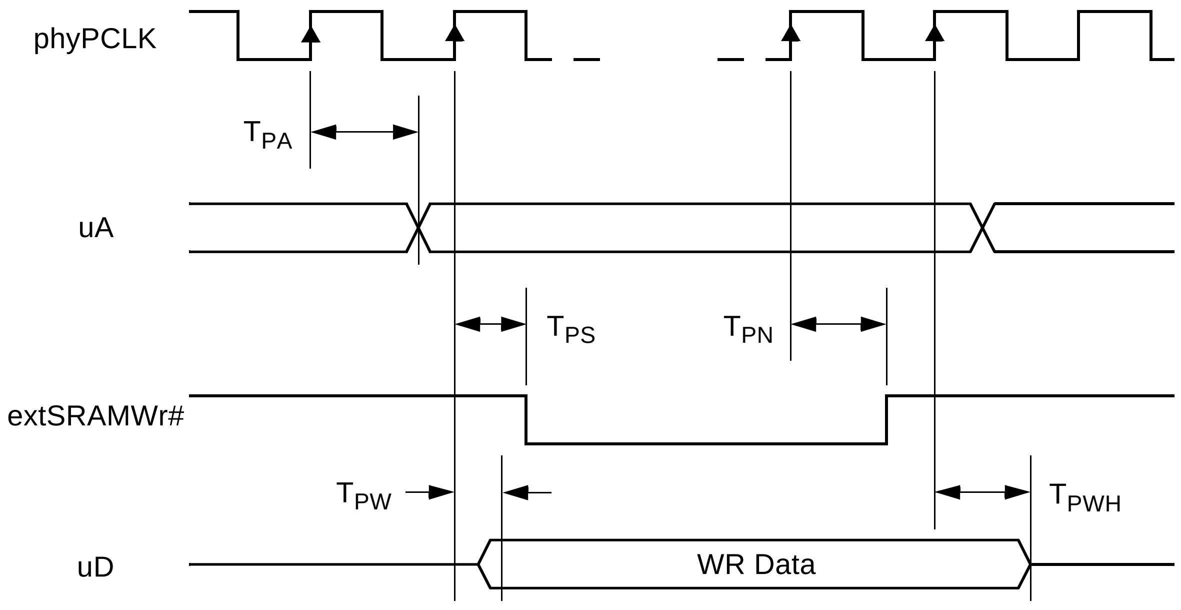


Figure 6-6 External SRAM Write Cycle

This page is intentionally left blank.

SECTION 7
Packaging Specifications

7.1 INIC-2430 LQFP Packaging Specifications

Figure 7-1 shows the physical outline of the 144-pin **LQFP** package. Table 7-1 shows the package's dimensions.

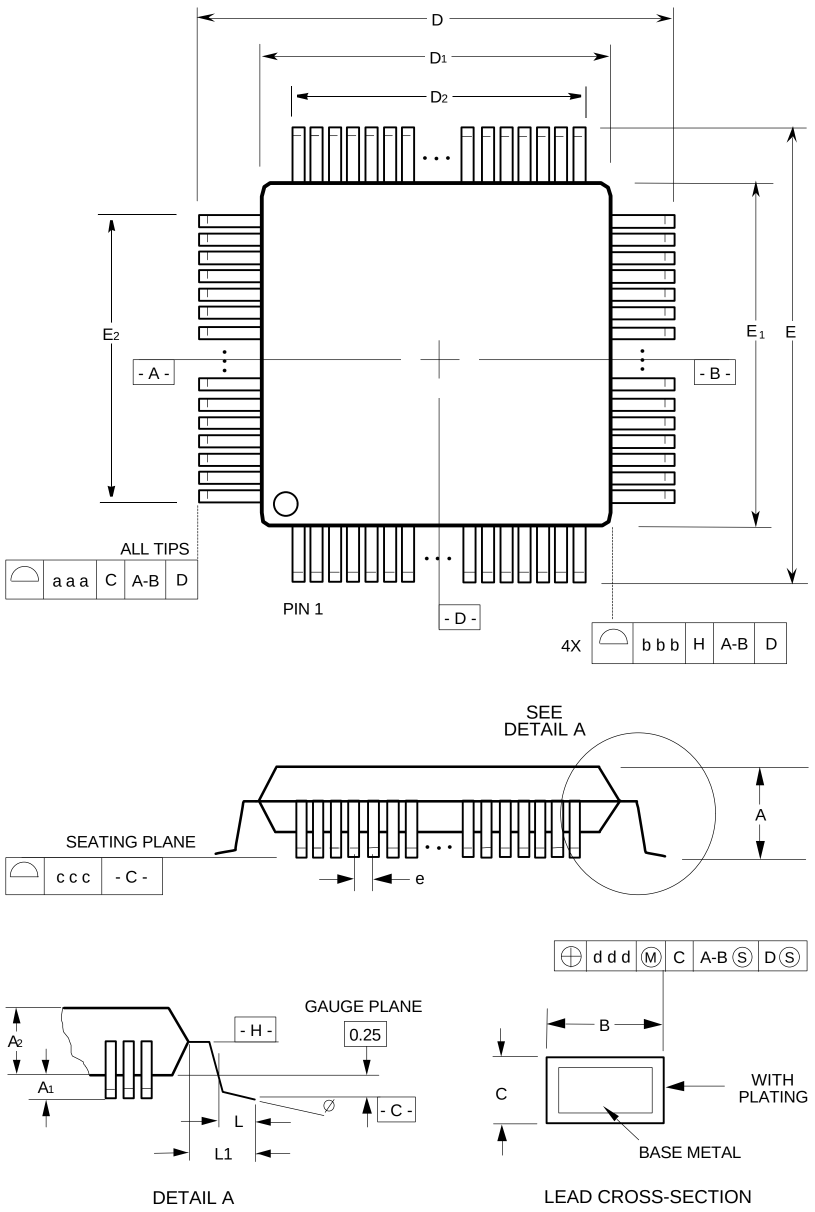


Figure 7-1 144 Pin LQFP Package Outline

Table 7-1 144-Pin LQFP Package Dimensions

SYMBOL	MM			INCH		
	MIN	NOM	MAX	MIN	NOM	MAX
A	- - -	- - -	1.60	- - -	- - -	0.063
A1	0.05	- - -	- - -	0.002	- - -	- - -
A2	1.35	1.40	1.45	0.053	0.055	0.057
B	0.17	0.22	0.27	0.007	0.009	0.011
C	0.12	- - -	0.20	0.005	- - -	0.008
D	21.85	22.00	22.15	0.860	0.866	0.872
D1	19.90	20.00	20.10	0.783	0.787	0.791
E	21.85	22.00	22.15	0.860	0.866	0.872
E1	19.90	20.00	20.10	0.783	0.787	0.791
e	0.50 BSC			0.020 BSC		
L	0.45	0.60	0.75	0.018	0.024	0.030
L1	1.00 BSC			0.039 BSC		
Ø	0 deg	3.5 deg	7 deg	0 deg	3.5 deg	7 deg
aaa	0.20			0.008		
bbb	0.20			0.008		
ccc	0.08			0.003		
ddd	0.07			0.003		

**NOTES:*

1. *Reference documents: JEDEC MS-026*
   Faraday dwg no. LQ144-20x20-01

2. *Controlling dimensions are in millimeters (mm).*

2. *The top package body size may be smaller than the bottom package body size by as much as 0.15 mm.*

3. *Datums A-B and -D- to be determined at datum plane -H-.*

4. *Reference plane -H- is located at mold parting line and is coincident with bottom of lead where it exits plastic body.*

5. *Dimensions D and E to be determined at seating plane -C-.*

6. *Dimensions D1 and E1 do not include mold protrusion. Allowable protrusion is 0.25 mm per side. Dimensions D1 and E1 are maximum plastic body size dimensions including mold mismatch.*

7. *Dimension B does not include dambar protrusion. Allowable dambar protrusion shall not cause the lead width to exceed the maximum B dimension by more than 0.08 mm. Dambar can not be located on the lower radius or the foot. Minimum space between protrusion and an adjacent lead is 0.07 mm.*

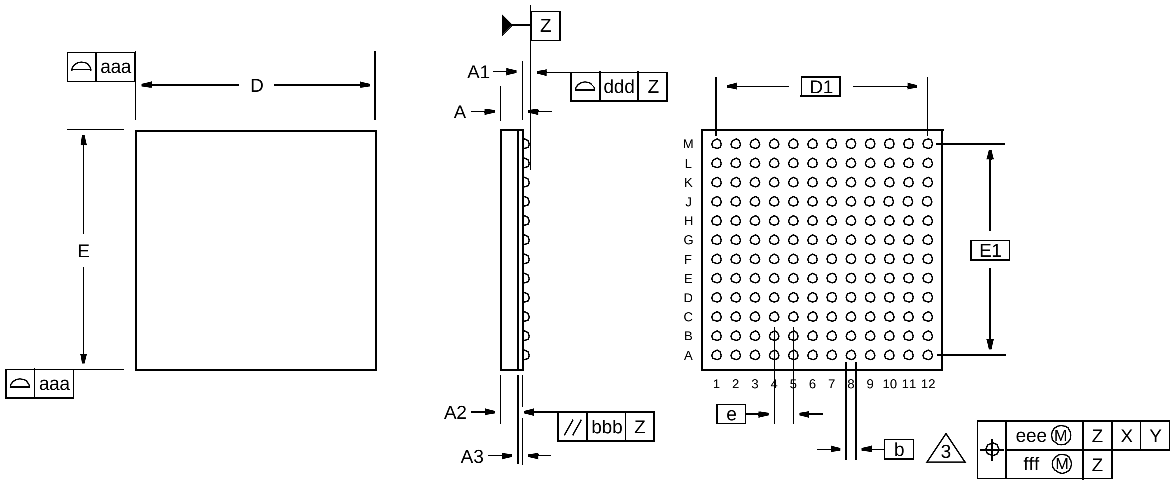
8. *The dimensions shown in lead cross-section apply to the flat section of the lead between 0.10 mm and 0.25 mm from the lead tip.*

9. *Dimension A1 is defined as the distance from the seating plane to the lowest point of the package body.*

10. *Solder plate thickness shall be 200 microinches minimum.*

7.2 INIC-2430 TFBGA Packaging Specifications

Figure 7-2 shows the physical outline of the 144-pin TFBGA package. Table 7-2 shows the package’s dimensions.



NOTES:

1. DIMENSIONING AND TOLERANCING PER ASME Y14.5M-1994.
2. ALL DIMENSIONS IN MM.
3. SOLDER BALLS ARE TYPICALLY 63/37 TIN LEAD.
4. REFERANCE DOCUMENT: JEDEC CODE MO-216.
5. PRIMARY DATUM Z AND SEATING PLANE ARE DEFINED BY THE SPHERICAL CROWNS OF THE SOLDER BALLS.
6. THE PATTERN OF PIN 1 FIDUCIAL IS FOR REFERENCE ONLY.
7. THERE SHALL BE A MINIMUM CLEARANCE OF 0.25mm BETWEEN THE EDGE OF THE SOLDER BALL AND THE BODY EDGE.
8. DIMENSION b IS MEASURED AT THE MAXIMUM SOLDER BALL DIAMETER, PARALLEL TO PRIMARY DATUM Z.

Figure 7-2 144-Pin TFBGA Package Outline

Table 7-2 144-Pin TFBGA Package Dimensions

	MM			INCH		
SYMBOL	MIN	NOM	MAX	MIN	NOM	MAX
A	- - -	- - -	1.30	- - -	- - -	0.051
A1	0.25	0.30	0.35	0.010	0.012	0.014
A2	0.84	0.89	0.94	0.033	0.035	0.037
A3	0.32	0.36	0.40	0.013	0.014	0.016
b	0.35	0.40	0.45	0.014	0.016	0.018
D	12.90	13.00	13.10	0.508	0.512	0.516
D1	- - -	11.00	- - -	- - -	0.433	- - -
E	12.90	13.00	13.10	0.508	0.512	0.516
E1	- - -	11.00	- - -	- - -	0.433	- - -
e	- - -	1.00	- - -	- - -	0.039	- - -
aaa	- - -	0.10	- - -	- - -	0.004	- - -
bbb	- - -	0.10	- - -	- - -	0.004	- - -
ddd	- - -	0.12	- - -	- - -	0.005	- - -
eee	- - -	0.15	- - -	- - -	0.006	- - -
fff	- - -	0.08	- - -	- - -	0.003	- - -

**NOTES:*

1. *Reference documents: JEDEC MO-216*
   Faraday dwg no. TF144-13x13-01
2. *Controlling dimensions are in millimeters (mm).*
3. *Dimension A is defined as the distance from the seating plane to the highest point of the package body.*