21X4 Serial ROM Format

Version 4.09
3-Mar-1999

Copyright (c) 1998-1999 Intel Corporation

This specification as well as the software described in it is furnished under license and may only be used or copied in accordance with the terms of the license. The information in this manual is furnished for informational use only, is subject to change without notice, and should not be construed as a commitment by Intel Corporation. Intel Corporation assumes no responsibility or liability for any errors or inaccuracies that may appear in this document or any software that may be provided in association with this document.

Except as permitted by such license, no part of this document may be reproduced, stored in a retrieval system, or transmitted in any form or by any means without the express written consent of Intel Corporation.

Modification History

DATE	*DOCUMENT VERSION*	*DETAILS*
31-Jan –1995	1.00	Initial Release
08-Mar-1995	1.01	This release corresponded to the first driver release supporting new SROM format. The wording associated with the EXT bit was changed and the documentation corrected to indicate that Media specific data for the DC21041 is 6 bytes (if used) and not 6 bits as previously stated. (The diagram had been accurate). The connection type codes for Full_Duplex were corrected: ■ Code 0x0204 replaces 0x0200 for the DC21041 & DC21140 ■ Code 0x0205 replaces 0x0203 for the DC21140 New Media codes were added: 0x04 – TP Full_Duplex was added for the DC21041 & DC21140 0x05 – SYM_SCR Full_Duplex (100BaseTx) was added for the DC21140 Rev 1.01 was also published as A.01.
30-Mar-1995		A recommendation to customers was added suggesting that they inform us of SROM areas that they are using for private data.
31-May-1995	1.02	Internal release only. The following Media codes for the DC21140 were added: 0x06 – 100BaseT4 0x07 – 100BaseFx (Fiber) 0x08 – 100BaseFx Full_Duplex The order of media in leaf was defined, thereby setting precedence for the autosensing algorithm (first being the lowest priority). The Enable_Autosense bit was added to the selected connection type. The Default_Media bit was added to the media code entry. The BNC media codes were added to the external documentation.
16-Jul-1995	1.03	The Manufacturer_Reserved field was added to the format. Appendix A, containing the SROM_CRC calculation algorithm, was added.
02-Aug-1995	1.03	The polarity of the Enable_Autosense bit was changed to Dynamic_Autosense_Disable (in the selected connection type).
06-Aug-1995	1.04	The Sub-system Vendor ID and Sub-system ID were added. Appendix B, containing the ID_BLOCK_CRC calculation algorithm, was added.

11-Aug-1995	3.00	The format was changed to include block types. Support for MII PHY chips was added, including media codes and PHY block information. NOTE: Version 2.00 was never released or implemented.
30-Nov-1995	3.00	Finalize version 3.00
22-Jan-1996	3.01	Support was added for 21142. The names were changed from DECchip 21xxx to Digital Semiconductor 21xxx and from DC21xxx to 21xxx.
08-Feb-1996	3.02	Fix the MediaCode value of 10BaseT Full_Duplex (0x04) in 21041's Media Block and 21142 Extended Format – Block Type 2.
27-Mar-1996	3.03	Add 21143 support. Block type 2 (in 21142 and 21143) describes SIA media, not all non-MII media.
13-May-1996	3.03	Add block type 5 – reset sequence
28-May-1996	3.03	Finalize version 3.03 Editorial changes, add warning regarding GPR sequences with no data.
28-Aug-1996	4.00	Support Magic Packet format. Modified App. B as well.
2-Sep-1996	4.00	Finalize version 4.00
24-Jul-1997	4.02	Support for Dual Function information. The format of the ID Block was changed to include dual function SROM format. Added Appendix C. Changed Wake-On-Lan to Magic Packet.
1-Dec-1997	4.03	Func0_HwOptions bit #5 (OnNowD3AuxPWR) changed to be reserved and must be zero.
25-Jan-1998	4.04	Support for Block Type 6. This block type applies to devices starting with 21143 Rev 4.1 and 21540.
1-Mar-1998	4.05	Fix incorrect description of the Length field of block type 5 and 6. Fix incorrect description of block type – what happens if more than one such block is present. (These changes are in the errata for version 4.04.)
10-Aug-1998	4.06	Added support for the 21145. Removed support for 21540 (no longer exists). This includes deleting the support for dual function, that was supported only by the 21540. Changed the name of the field "MII Connector Interrupt" in block type 3 to be "MII PHY Insertion/removal Indication" , to indicate the functionality of this field. Changed the MII connector interrupt possible values in block type 3, to support also a pluggable MII PHY, which doesn't generate an interrupt. Changed Func0_HwOptions <0:1> to be zero, which indicates that the EXPANSION ROM size is 256K for a double latch support - which is the default – and 512 Bytes for a single latch support. All other options are ignored, because the 21540 is no longer supported.
1-Oct-1998	4.07	Added back the support for dual function (Ethernet + Modem), since modem functionality was added to the 21145.
14-Jan-1999	4.08	Func0_HwOptions bit #5 must be one in 21145 Func0_HwOptions bit #2 must be zero for 21145. Single latch not supported. Update the table in Func0_HwOptions bit #0:1 to reflect all supported expansion ROM sizes.

3-Mar-1999	4.09	Support for Block Type 7. This block type applies to device 21145, and specifies the values that can be written to the HomeRun registers. These values will overwrite the driver default values.

Magic Packet is a trademark of Advanced Micro Devices, Inc.

1. OVERVIEW **7**

2. GENERAL SROM FORMAT **8**

3. SROM BOARD INFORMATION **10**

4. MAGIC PACKET INFORMATION BLOCK **13**

5. SINGLE FUNCTION ID BLOCK **16**

6. DUAL FUNCTION ID BLOCK **17**

6.1 FUNC1_HWOPTIONS (OFFSET 14D) 19
6.2 MISCHWOPTIONS (OFFSET 15D) 20
6.3 FUNC0_HWOPTIONS (OFFSET 17D) 21

7. CONTROLLER INFO LEAF **23**

7.1 INTRODUCTION 23
7.2 21041 INFO LEAF 24
7.2.1 21041 Info Leaf Format *24*
7.2.2 21041 Media Block Format *25*
7.3 21140 INFO LEAF 26
7.3.1 21140 Info Leaf Format *26*
7.3.2 21140 Info Block Format *28*
7.4 21142 INFO LEAF 35
7.4.1 21142 Info Leaf Format *35*
7.4.2 21142 Info Block Format *37*
7.5 21143 INFO LEAF 43
7.5.1 21143 Info Leaf Format *43*
7.5.2 21143 Info Block Format *45*
7.6 21145 INFO LEAF 51
7.6.1 21145 Info Leaf Format *51*
7.6.2 21145 Info Block Format *53*
APPENDIX A – SROM_CRC CALCULATION ALGORITHM 57
APPENDIX B – ID_BLOCK_CRC CALCULATION ALGORITHM 59
APPENDIX C – CIS POINTER SETTING FROM SROM 60
CIS DATA STORED IN SROM (ADDRESS SPACE INDICATOR = 2) 61
CALCULATING THE VALUE OF THE CIS POINTER FIELD FOR CIS IN SROM 62
CIS DATA STORED IN EXPANSION ROM (ADDRESS SPACE INDICATOR= 7) 63

1. Overview

The Serial ROM supports the following Ethernet controllers:

- 21041
- 21140
- 21140A
- 21142
- 21143
- 21145

The Serial ROM format is defined to provide support for:

- Multiple controllers on a single board sharing a single serial ROM
- Multiple MII PHY chips connected to the same controller.
- Dual functionality (Ethernet + Modem) on the same controller.

The fields within the SROM are parsed by the hardware and the driver, and give specific information related to the design.

It is important to note that the drivers that ship with many of the major OS's will use the supplied Intel driver for Plug-and-Play compatability, and that omission of a correctly formatted SROM may prevent the Plug-and-Play supplied driver from loading.

If manufacturers wish to ensure that their specific driver loads for their specific product, then they should make provisions to supply directions explaining how to ensure that.

NOTE:

1. To optimize the ROM space utilization, byte fields are also used. Since the serial ROM only supports word accesses, it is recommended that the entire ROM be first downloaded into a memory shadow table before be parsed.

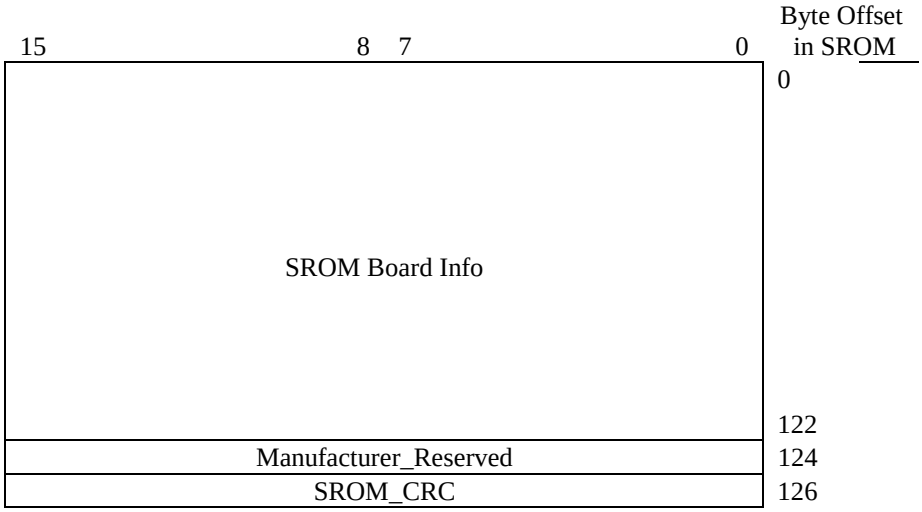
2. Those fields referred to throughout this specification as 'Reserved' must contain zeros (bytes and bits).

3. This specification permits board manufacturers to use parts of the SROM for private data. We advise that such usage is discussed with Intel, to help avoid conflicts with future versions of the SROM format. Please contact your local Intel representative for further information.

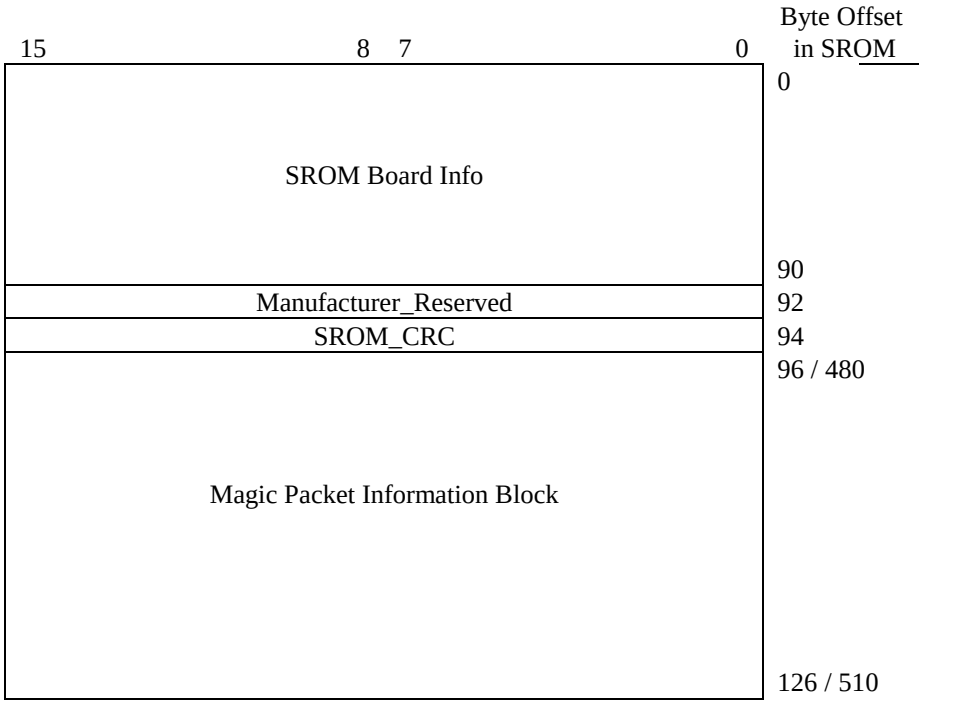
2. General SROM Format

The SROM can be in one of the following formats, depending on whether there is information for Magic Packet. See next sections for details of SROM board Info and Magic Packet Info.

If no Magic Packet information is present, the format is:

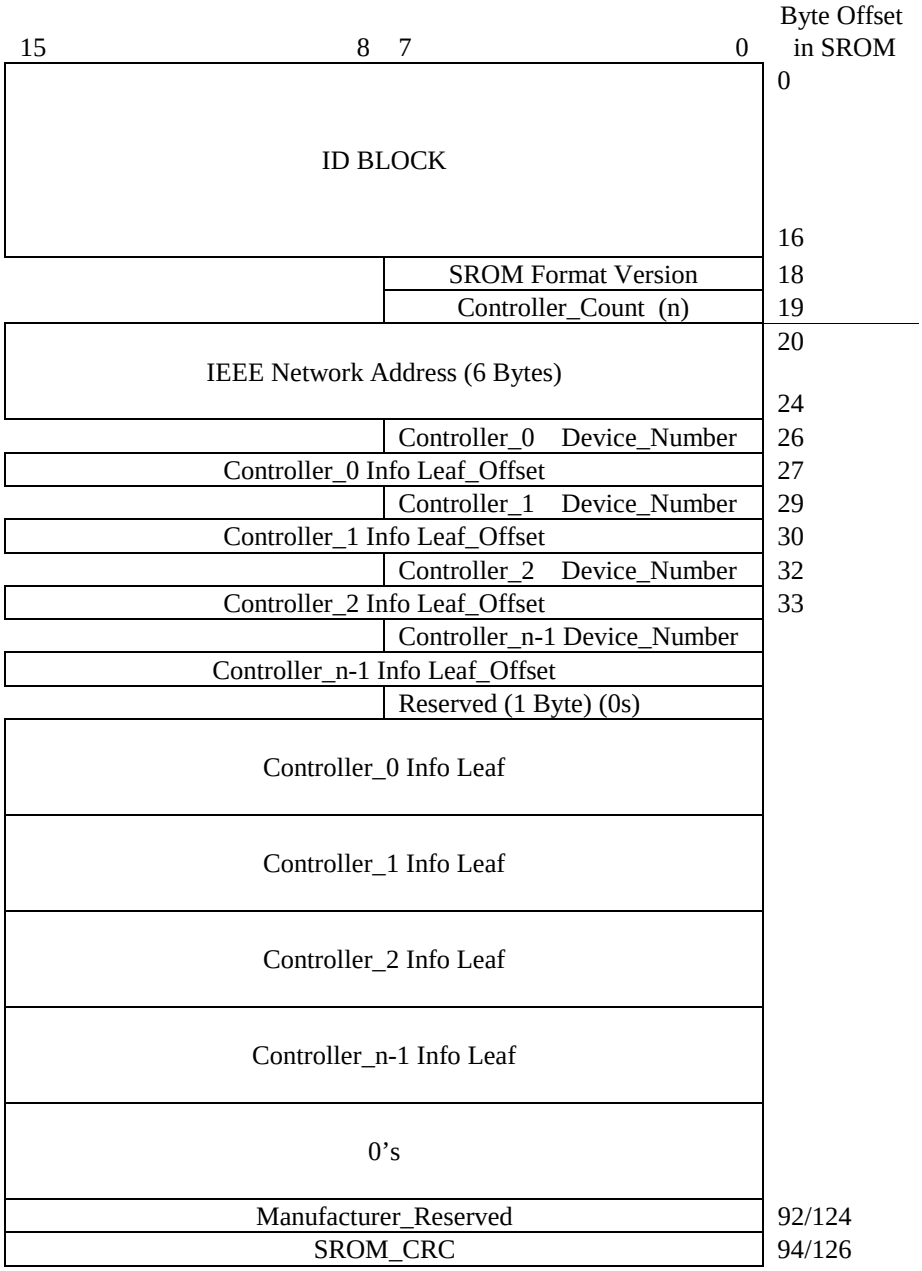


If Magic Packet information is present, the format is:



NOTE: Magic Packet information block is supported only by the 21143 Rev 2.0 and above and by the 21145. The 21145 does not support the SecureON password though.

3. SROM Board Information



Field	Size (Bytes)	Description
ID Block	18	Refer to section 5 below.
SROM Format Version	1	This field displays the version number of the SROM format. The current version is 0x04
Controller Count	1	This field displays the number of controllers sharing this ROM. A single port board will have a value of 1 in this field.
IEEE Network Address	6	This is the IEEE address of the controllers on a single controller board. On a multiple controller board, this is the base IEEE address. Every controller (0..n) adds its index (n) to this base IEEE address.
Controller_n Device_Number	1	There is one such field per controller sharing the SROM. On a multi-controller board, this field contains the DEVICE_NUMBER value. This is the value by which the configuration space of the n'th controller can be accessed on the board's secondary PCI bus. This value depends on the hardware routing of the board. The DEVICE_NUMBER is the 'chip select' line routed from the controller to the PCI-to-PCI bridge chip on board. On a single controller board this field has no meaning and should be ignored by the driver.
Controller_n Info Leaf_Offset	2	This is the Byte offset (from the beginning of SROM) where the controller_n info block is located. There is one such field per controller sharing the SROM. The information block is controller specific i.e. it varies between controllers. See the format of the controller leaves (Section 6) for details. NOTE: If multiple controllers have identical information blocks, a single leaf can be shared and all leaf pointers can be set to point to that leaf. This is correct only if the user cannot choose between multiple media ports for each controller. (See Section 6 for details.) For example: a 4_TP port card can share one info block for all four controllers.

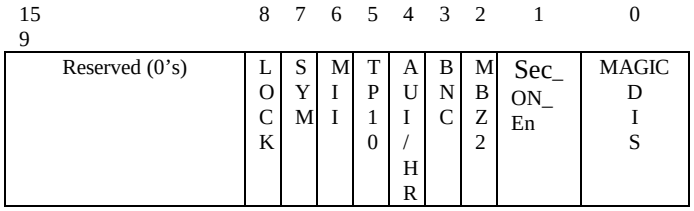
ASCIIZ Sign On Message	L	ASCII string, terminated by a zero byte. This message will be printed when the driver has loaded, following the standard Intel message (currently not implemented). Note that the location of this field depends on the number of controllers supported by this card. The length of this field depends on the space remaining in the SROM
Manufacturer_ Reserved	2	This field is reserved for the specific use of the controller manufacturer. Drivers developed by Intel will not make any use of this field. The field will always be located in the two bytes which precede the SROM_CRC. If the manufacturer's private data is more than two bytes, the field can be used as a pointer to the actual data. If not used, this field must be filled with 0's (MBZ). This field is located at byte offset 124 in the SROM if no Magic Packet information is present, and at offset 92 if the Magic Packet information block is present.
SROM_CRC	2	The value in this field is determined by all the words of the SROM from word[0] to the word before the CRC i.e. (word[SROM_word_size - 2]). The CRC word is derived by calculating the CRC32 of all the SROM until (and not including) the last word, and taking the two least significant bytes of the result. The bytes are written in little endien. The function definition is contained in Appendix A of this document. This field is located at byte offset 126 in the SROM if no Magic Packet information is present, and at offset 94 if the Magic Packet information block is present.

4. Magic Packet information block

15 8	7 0	Byte Offset in SROM	
ID BLOCK		0 16	
	SROM Format Version	18	
	Controller_Count (n)	19	
IEEE Network Address (6 Bytes)		20 24	
See SROM Board Information, Section 3, for details			
Manufacturer_Reserved		92	
SROM_CRC (calculated on bytes 0:95)		94	
* SecureON Password (B)	* SecureON Password (A)	96/480	Magic Packet Block
* SecureON Password (D)	* SecureON Password (C)		
* SecureON Password (F)	* SecureON Password (E)	100	
Magic's IEEE address (B)	Magic's IEEE address LSB (A)	102	
Magic's IEEE address (D)	Magic's IEEE address (C)	104	
Magic's IEEE address MSB (F)	Magic's IEEE address (E)	106	
Magic Command (MSB)	Magic Command (LSB)	108	
Magic_Reserved1 (16 Bytes) 0's		110 124	
Magic_Reserved2 (0)	MAGIC_BLOCK_CRC (calculated on bytes 96:127)	126/510	

1K SROM / 4K SROM

* NOTE: The SecureON Password field is not supported by the 21145.

<table>
<tr><th>Field</th><th>Size (Bytes)</th><th>Description</th></tr>
<tr><td>SecureON Password</td><td>6</td><td>This field contains the SecureON password. If the SecureON feature is enabled, only remote Magic packets that have valid passwords will invoke the system. This field is enabled by setting bit 1 in the remote Magic command word in the SROM or the equivalent bit in the remote Magic command register.
If SecureON is disabled this field must be zero.
The SecureON password uses the naming convention of A-B-C-D-E-F, with “A” representing the first byte of the SecureON password.
This field must be zero for the 21145.</td></tr>
<tr><td>Magic’s IEEE Address</td><td>6</td><td>This field contains the Magic’s IEEE address on which the workstation should be woken up. This field’s format is the same as the IEEE Network Address format. Only Magic packets that passed Magic’s IEEE Address filtering will invoke the system (note that also broadcast packets pass address filtering).
The Magic IEEE address uses the naming convention of A-B-C-D-E-F, with “A” representing the first byte of the Magic IEEE address.</td></tr>
<tr><td>Magic Command</td><td>2</td><td>Controls Magic/Power Management functionality.
Bits #3-7 define the media on which the chip will attempt to autosense while in magic mode, or in D1, D2 or D3 states.
 

MAGIC_DIS When set to 1, disables the Magic functionality.
This bit should be programmed to 0 for the 21143 Rev 4.1.
Sec_ON_En Enables SecureON password matching mode and functionality.
Must be 0 for the 21145.
MBZ2 This bit must be 0.
BNC If set to 1, autosense will be attempted also on BNC.
This bit should be programmed to 0 for HomeRun.
AUI/HR In the 21145, if set to 1, autosense will be attempted to autodetect also on HomeRun.
For all other devices supporting magic block, if set to 1, autosense will be attempted also on AUI.
TP10 If set to 1, autosense will be attempted also on TP 10Mbps.</td></tr>
</table>

		MII	If set to 1, autosense will be attempted also on MII PHY's port.
		SYM	If set to 1, autosense will be attempted also on symbol PHY's port.
		LOCK	If set to 1, Magic Packet's Command register is locked for further writes/reads. If locked, this register reads all 1's. If locked, Magic Packet's IEEE address register is not writeable. Chip exits from LOCK state only in HW reset.
		Reserved (7 bits)	These bits must be 0's.
Magic_Reserved1	16	Must be Zeros	
MAGIC_BLOCK_CRC	1	This field contains the CRC8 of the Magic block, calculated on byte[96] ... byte[127] (inclusive) in a 1K SROM, or on byte[480]…byte[511] (inclusive) in a 4K SROM. Please note that this also includes the Magic_Reserved2 field. The function definition is contained in Appendix B of this document.	
Magic_Reserved2	1	Must be Zero.	

5. Single Function ID Block

This format should be used for boards that support only the Ethernet function.

15 8	7 0	Byte Offset in SROM	
Sub-System Vendor ID		0	
Sub-System ID		2	I
Cardbus CIS Pointer low		4	D
Cardbus CIS Pointer high		6	
ID_Reserved1 (0s) (7 bytes)		8	B L O
MiscHwOptions		14	C
Func0_HwOptions	ID_BLOCK_CRC	16	K

Field	Size (Bytes)	Description
Sub-System Vendor ID	2	This field contains the Sub-System Vendor ID of this controller. The data in this field is used to uniquely identify the controller in order to distinguish it from other controllers based on the 21x4 family of chips. Refer to the PCI specification.
Sub-System ID	2	This field contains the Sub-System ID of this controller. The data in this field is used to uniquely identify the controller to distinguish it from other controllers based on the 21x4 family of chips. Refer to the PCI specification.
CardBus CIS Pointer Low	2	This field determines the tuples location in the BROM or in the SROM. It is used to calculate CardBus CIS Pointer Low field. The CIS Pointer calculation is specified in Appendix C of this document.
CardBUS CIS Pointer High	2	This field determines the tuples location in the BROM or in the SROM. It is used to calculate CardBus CIS Pointer High field. The CIS Pointer calculation is specified in Appendix C of this document.
ID_Reserved1	7	Must be 0s.
MiscHwOptions	1	This field sets hardware parameters. The field's details are described in section 6.1 below.
Func0_ HwOptions	1	This field sets the EXPANSION ROM, PME, and CLKRUN configuration. The field's details are described in section 6.2 below.

6. Dual function ID Block

This format should be used for boards that support both an Ethernet function and a modem function.

15	8	7	0	Byte Offset in SROM	
Sub-System Vendor ID - Ethernet & Modem				0	
Sub-System ID - Ethernet				2	I
CardBus CIS Pointer low - Ethernet & Modem				4	D
CardBus CIS Pointer high - Ethernet & Modem				6	
Reserved0 (0s)		Sub-System ID - Modem		8	B
Modem Class Code (Sub-Class)		Modem Class Code (Interface)		10	L
Reserved1 (0s)		Reserved2 (0s)		12	O
MiscHwOptions		Func1_HwOptions		14	C
Func0_HwOptions		ID_BLOCK_CRC		16	K

Field	Size (Bytes)	Description
Sub-System Vendor ID – Ethernet & Modem	2	This field contains the Sub-System Vendor ID of this controller. The data in this field is used to uniquely identify the controller in order to distinguish it from other controllers based on the 21x4 family of chips. This field is common to both the Ethernet and the modem functions. Refer to the PCI specification.
Sub-System ID – Ethernet	2	This field contains the Sub-System ID of this Ethernet controller. The data in this field is used to uniquely identify the Ethernet controller to distinguish it from other controllers based on the 21x4 family of chips. The high byte of this Sub System ID of the Ethernet is identical to the high byte of the Sub system ID of the Modem. Refer to the PCI specification.
CardBus CIS Pointer Low – Ethernet & Modem	2	This field is common to the Ethernet and to the Modem functions. It determines the tuples location in the BROM or in the SROM. It is used to calculate both functions' CardBus CIS Pointer Low fields. The CIS Pointer calculation is specified in Appendix C.
CardBus CIS Pointer High – Ethernet & Modem	2	This field is common to the Ethernet and to the Modem functions. It determines the tuples location in the BROM or in the SROM. It is used to calculate both functions' CardBus CIS Pointer High fields. The CIS Pointer calculation is specified in Appendix C.

Sub-System ID – Modem	1	This field defines the Low byte of the Modem's Sub-System ID. The data in this field is used to uniquely identify the modem controller to distinguish it from other controllers based on the 21x4 family of chips. The Modem's Sub-System ID High byte is identical to the Ethernet Sub-System ID High byte (refer to the Sub System ID of the Ethernet above)..
Reserved0	1	Must be 0s.
Modem Class Code (Interface)	1	This field defines the Low byte of the Modem's Class-Code 3 Bytes field. The Modem's Class Code High byte (Base Class) is hardcoded to a value of 7Hex (for BROM tuples), or 2 Hex (for SROM tuples).
Modem Class Code (Sub-Class)	1	This field defines the Mid byte of the Modem's Class-Code 3 Bytes field. The Modem's Class Code High byte (Base Class) is hardcoded to a value of 7Hex (for BROM tuples), or 2 Hex (for SROM tuples).
Reserved2	1	Must be 0s.
Reserved1	1	Must be 0s.
Func1_ HwOptions	1	This field sets the modem parameters. It is meaningful only when a modem is supported. Otherwise This field must be 0's. The field's details are described in section 6.1.
MiscHwOptions	1	This field sets hardware parameters. The field's details are described in section 6.2.
Func0_ HwOptions	1	This field sets the Ethernet parameters. It is meaningful only when the tuples are located in the EXPANSION ROM. Otherwise this field must be 0's. The field's details are described in section 6.3 below.
ID_BLOCK_CRC	1	This field contains the CRC8 of the ID block, calculated on word[0] … word[8] (inclusive). Please note that this also includes the ID_Reserved2 field. The function definition is contained in Appendix B of this document.

6.1 Func1_HwOptions (offset 14d)

<table>
<tr><th>bit</th><th>Bit Name</th><th>Description</th></tr>
<tr><td>0</td><td>ModemEnable</td><td>If set to 1, Modem function is enabled. Otherwise, Modem function is disabled.</td></tr>
<tr><td>2:1</td><td>NumModemRegs</td><td>These bits declare the number of Modem registers. The CBIO decoding for modem is determined by these bits.

<table><tr><th>Number of Modem regs</th><th>SROM code</th></tr><tr><td>8</td><td>00</td></tr><tr><td>16</td><td>01</td></tr><tr><td>32</td><td>10</td></tr><tr><td>Reserved</td><td>11</td></tr></table></td></tr>
<tr><td>3</td><td>IgnoreRdyOnFirst Access</td><td>This bit determines the MDM_WAIT signal pin.
When this bit is set, the MDM_WAIT pin is used as ISA IOCHRDY through which the modem can insert wait states in modem chipset read and write transactions.
In this mode, the 21145 ignores the MDM_WAIT pin before starting the transaction and examines its value only during the transaction.

When this bit is cleared, the 21145 examines the MDM_WAIT pin after modem reset. The 21145 will not start any transaction to the modem chipset before the MDM_WAIT pin is deasserted.
When using this mode, the MDM_WAIT pin is also used as ISA IOCHRDY.</td></tr>
<tr><td>4</td><td>RI_Polarity</td><td>Modem Ring Indicator input polarity.
0 – RI_L assumed.
1 – RI_H assumed</td></tr>
<tr><td>5</td><td>PwrControlPolarity</td><td>Modem Power Control output line polarity
0 – POWER_L assumed
1 – POWER_H assumed</td></tr>
<tr><td>6</td><td>Reserved</td><td>Must be zero.</td></tr>
<tr><td>7</td><td>ModemAudioSelect</td><td>1 – Modem Function Event Mask register bit5 is supported (BAM – binary audio enable/disable).
0 – Modem Function Event Mask register bit6 is supported (PWM – PWM audio enable/disable).</td></tr>
</table>

6.2 MiscHwOptions (offset 15d)

bit	Bit Name	Description
0	Gep3LedDefinition	This bit controls the Ethernet GEP3 pin behavior when it is used as a LED output pin (programmed by SW through CSR15<23>). When GEP3 functions as a LED, this bit has the following meaning: 0 – GEP3 LED shows the LINK. 1 – GEP3 LED shows the LINK + Activity.
1	PME_STSCHG	This bit controls the PME_L/STSCHG pin polarity. The polarity and the meaning of the signal are different in PCI and in CardBus. 0 – Active Low (PME-L) 1 – Active High (STSCHG)
7:2	Reserved	Must be zero.

6.3 Func0_HwOptions (offset 17d)

<table>
<tr><th>bit</th><th>Bit Name</th><th>Description</th></tr>
<tr><td>1:0</td><td>Reserved
(BromSize)</td><td>These bits should be set according to the EXPANSION ROM size.
According to this, the dual function 21145 will split the EXPANSION ROM for CardBus applications that need Tuples for two functions:

<table><tr><th>Expansion ROM size (Single BROM Latch ==1)</th><th>Expansion ROM size (Single BROM Latch == 0)</th><th>SROM code</th></tr><tr><td>Reserved</td><td>4k</td><td>11</td></tr><tr><td>Reserved</td><td>16k</td><td>10</td></tr><tr><td>256 Bytes</td><td>64k</td><td>01</td></tr><tr><td>512 Bytes</td><td>256k</td><td>00</td></tr></table></td></tr>
<tr><td>2</td><td>SingleBromLatch</td><td>If set to 1, BOOT ROM interface assumes one latch implementation. Otherwise, two latches implementation is assumed.
For 21145 SingleBromLatck is not longer supported, so this bit must be zero.</td></tr>
<tr><td>3</td><td>PME_Enable</td><td>If set to 1, PME functionality is enabled (GEP2 pin is used as PME signal or as STSCHG signal). Otherwise, backward compatibility is maintained (GEP2 pin is used as a GEP pin). This bit is the New Capabilities bit (bit 4) of PCI status register of the Ethernet and of the Modem function.
The New Capabilities bit of the PCI status register is set or cleared according to this bit.</td></tr>
<tr><td>4</td><td>EnableCLKRUN</td><td>Controls the functionality of CLKRUN pin.
0 – Simplified CLKRUN support. While in this mode, the controller (21143 Rev 4.1, or 21145) will always refuse to slow the clock speed or to stop it.
1 – Enhanced CLKRUN support. While in this mode, the controller (21143 Rev 4.1, or 21145) will agree to slow or stop the clock when it does not transmit or receive. This mode also improves the power consumption by that it supports
deep sleep/snooze mode.</td></tr>
<tr><td>5</td><td>Reserved</td><td>Must be zero. In Luxor must be one.</td></tr>
</table>

6	OnNowD3ColdCap	On Now D3 state capability. This bit defines D3cold support. This bit affects **PME_Support_D3cold** bit in the CCID register. Using this bit is implementation specific. In order to support D3Cold, the design must support an auxiliary power connection. (The auxiliary power source can be in the system or on the board.)
7	RealSTSCHG	If set to 1, STSCHG registers (relevant in CardBus only) will effect interrupts and power management behavior.

7. CONTROLLER INFO LEAF

7.1 Introduction

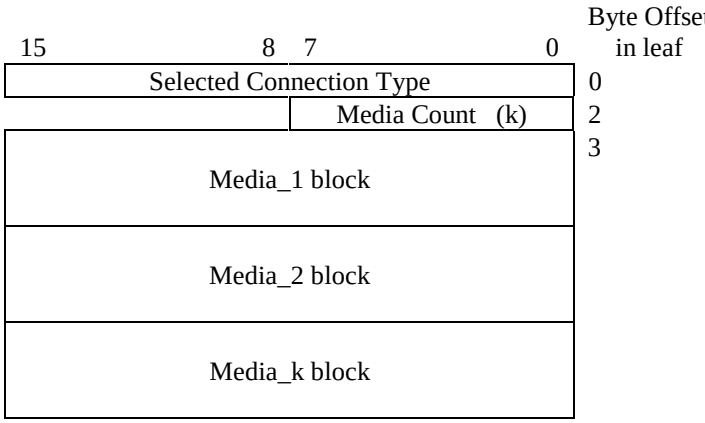
The controller info leaf is the structure that contains all the board specific data. This information differs between controllers, because of the different functionality each offers. While the SROM ID block described in the previous section is common to all devices, the format of the info leaves for the various devices differs.

The following table summarizes the various formats associated with each controller:

Block Format	21041	21140/A	21142	21143	21145
21041 Block Format	✓	–	–	–	–
21140 Compact	–	Non MII	–	–	–
Extended Type 0	–	Non MII	–	–	–
Extended Type 1	–	MII	–	–	–
Extended Type 2	–	–	SIA Media	SIA Media	SIA Media
Extended Type 3	–	–	MII	MII	MII
Extended Type 4	–	–	–	SYM Media	SYM Media
Extended Type 5	–	✓	✓	✓	✓
Extended Type 6	–	–	–	✓	✓
Extended Type 7	–	–	–	–	✓

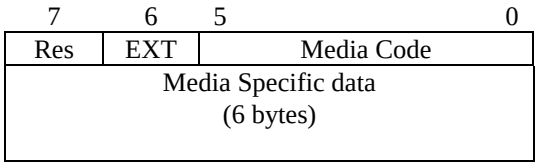
7.2 21041 INFO LEAF

7.2.1 21041 Info Leaf Format



Field	Size (Bytes)	Description
Selected Connection Type	2	The connection type utilized by the adapter is usually selected by the user in the drivers' configuration files. The purpose of this field is to allow information that cannot be modified by the setup utilities in the configuration files to be saved in the SROM instead. The possible values are: 0x0000 10BaseT (Twisted Pair) 0x0100 10BaseT with Nway 0x0204 10BaseT Full_Duplex 0x0400 10BaseT without Link Pass test 0x0001 10Base2 (BNC) 0x0002 10Base5 (AUI) 0x0800 AutoSense 0x0900 AutoSense with Nway Enabled If this field is not used, it must be set to 0xFFFF. Any other value is invalid and may cause unpredictable results.
Media Count	1	This field specifies the number of media blocks present for this controller.
Media_k block	Media Dependent	The data in this field describes a particular supported medium. There is one such field per supported medium. See Section 7.2.2 for further details.

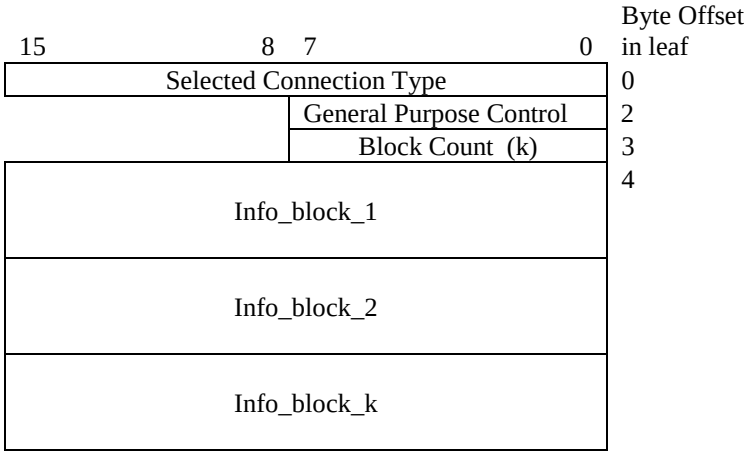
7.2.2 21041 Media Block Format



Field	Size (Bits)	Description
Media Code	6	This indicates to the driver that this medium is supported by the controller. Possible values are: 0x00 10BaseT (Twisted Pair) 0x01 10Base2 (BNC) 0x02 10Base5 (AUI) 0x04 10BaseT Full_Duplex
EXT	1	When set to 1, this field indicates that the lower 16 bits of CSRs 13-15 (i.e. the SIA registers) will be set via the SROM and are not using the internal default values for this media type. The SIA setting values are provided by the Media Specific Data field. This bit exists only for the 21041.
Res	1	Reserved
Media Specific Data	48	This field contains valid data only when the EXT bit is set. This field provides the values of CSR13, CSR14, CSR15 that should override the driver internal defaults for this media type. 15 8 | 7 0 CSR13<15..0> CSR14<15..0> CSR15<15..0>

7.3 21140 Info Leaf

7.3.1 21140 Info Leaf Format



Field	Size (Bytes)	Description
Selected Connection Type	2	The connection type utilized by the controller is usually selected by the user in the drivers' configuration files. The purpose of this field is to allow information that cannot be modified by the setup utilities in the configuration files to be saved in the SROM instead. The standard case (where the media selection information is stored in the drivers' configuration files) is for this field to be set to one of the following values, depending on the board's capabilities: 0x0800 : Powerup AutoSense and Dynamic AutoSense if the board supports it. 0x8800 : Powerup AutoSense only

		For those setup utilities using the SROM, the possible values are: 0x0000 10BaseT (Twisted Pair) 0x0100 10BaseT with Nway 0x0204 10BaseT Full_Duplex 0x0001 10Base2 (BNC) 0x0002 10Base5 (AUI) 0x0003 100BaseTx (Symbol Scrambler) 0x0205 100BaseTx Full_Duplex 0x0006 100BaseT4 0x0007 100BaseFx (Fiber) 0x0208 100BaseFx Full_Duplex 0x0009 MII 10BaseT 0x020A MII 10BaseT Full_Duplex 0x000D MII 100BaseTx 0x020E MII 100BaseTx Full_Duplex 0x000F MII 100BaseT4 0x0010 MII 100BaseFx 0x0211 MII 100BaseFx Full_Duplex 0x0800 Powerup AutoSense, Dynamic AutoSense 0x8800 Powerup AutoSense only 0xFFFF No selected media information Any other value is invalid and may cause unpredictable results.
General Purpose Control	1	This field contains the value of the General Purpose mask register of controller_n, regardless of the media involved. This value is controller specific. It determines the direction of the General Purpose port bits (i.e. those bits that will be inputs and those that will be outputs).
Block Count	1	This value represents the number of info blocks present for this controller.
Info_k_block	Media Dependent	Each of these fields describes a particular supported medium/PHY chip. There is one such field for every supported non-MII medium and one for every MII PHY chip. See Section 7.3.2 for further details. The order of the info blocks defines their precedence during autosensing: The first entry is the medium/PHY chip of lowest precedence and is the one that will be checked last. The final entry in the list is the medium/PHY chip with highest precedence and is checked first.

7.3.2 21140 Info Block Format

The Info block can be found in either of the following formats:

- Compact format (as per format version 1.04) - for non-MII media only
- Extended format - for both MII and non-MII media, and for controller general information (reset info, for example).

The two blocks can be distinguished by bit 7 of the first byte of the block:

"0" indicates that the info block is in compact format,
"1" indicates that the info block is in extended format.

7.3.2.1 21140 Compact Format

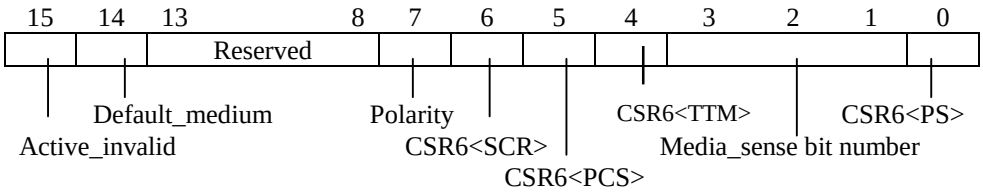
This info block should only be used to describe non-MII media.

7	6	5 ... 0
F(0)	RES	Media Code
General Purpose Port Data		
Command (2 Bytes)		

Field	Size (Bits)	Description
Media Code	6	This field indicates to the driver that this particular medium is supported by the controller. Possible values are: 0x00 10BaseT (Twisted Pair) 0x01 10Base2 (BNC) 0x02 10Base5 (AUI) 0x03 100BaseTx (Symbol Scrambler) 0x04 10BaseT Full_Duplex 0x05 100BaseTx Full_Duplex 0x06 100BaseT4 0x07 100BaseFx (Fiber) 0x08 100BaseFx Full_Duplex
Reserved	1	Reserved.
Format Indicator	1	The value in this field must be zero for compact format.
General Purpose Port Data	8	When this medium is selected, these 8 data bits are written to the General Purpose data register of controller_n (21140). The value of this parameter is board and controller specific. The data is defined by the board manufacturer, and its purpose is to initialize and enable the selected medium's hardware

Command	16	When this medium is selected, this field is used for generating the CSR6's mode bits of controller_n (21140). It is also used to obtain the bit number in the General Purpose port register from which the driver senses the media activity (bits <3:1>).

A description of the command field now follows.

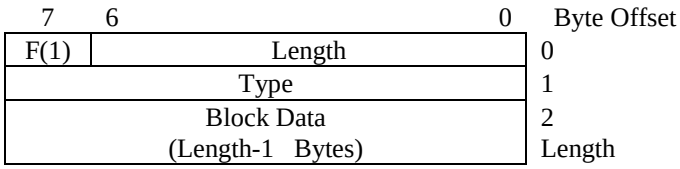


Active_invalid	If set, this bit indicates that the Media Sense bit number is not valid and that there is no media activity indication in the General Purpose Register.
Default_medium	Dynamic Autosensing will only be attempted between media for whom this bit is clear (indicating that there is a valid media sense bit to test). If set, this bit indicates the default medium that will be selected if no active link is found during the autosense process (Powerup and Dynamic). This bit is valid only if the Active_invalid bit is clear for this medium. This bit is not valid for Full_Duplex media entries. It should be set for one medium only.
Polarity	This bit indicates the polarity of the media activity indication bit in the General Purpose Register (meaningful only if the Active_invalid bit is not set). When this bit is clear, the media activity bit in the GPR is active-high i.e. it will read '1' when the medium is active. When this bit is set, the media activity bit in the GPR is active-low i.e. it will read '0' when the medium is active. The following table describes the activity indication:

GPR Activity Bit	Polarity Bit	Indicates the Medium is:
0	1	Not active
0	1	Active
1	0	Active
1	1	Not active

7.3.2.2 21140 Extended Format

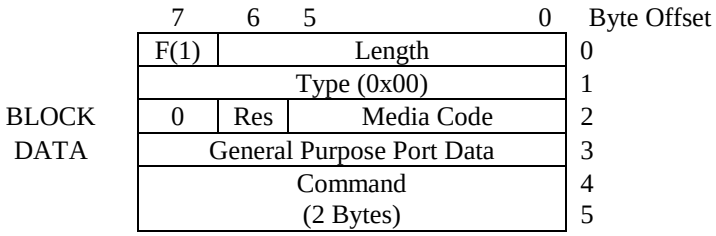
This info block can be used to describe both non-MII media and MII PHY chips.



Field	Size (Bits)	Description
Length	7	This field contains the size, in bytes, of this info block. The size includes the type field and the block data but does NOT include the length field itself.
Format Indicator	1	The value in this field must be 1 for Extended Format.
Type	8	This represents the type of Extended block. The 21140 currently supports: 0x00 non-MII media block 0x01 MII PHY media block.
Block data	8* (Length-1)	The value in this field is determined by the block type. See Sections 7.3.2.2.1 and 7.3.2.2.2.

7.3.2.2.1 21140 Extended Format – Block Type 0

This block type can be used to describe non-MII media only. The data in this block is identical to the Compact info block as described above in Section 7.3.2.1.



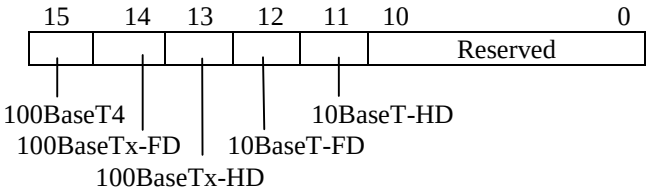
Field	Size (Bits)	Description
Length	7	The value in this field is always 0x05 for Block Type 0.
Format Indicator	1	The value in this field must be 1 for Extended Format.
Type	8	This field displays the Block Type: 0x00 .
Block Data	32	See the Compact Info Block described in Section 4.2.2.1 for details.

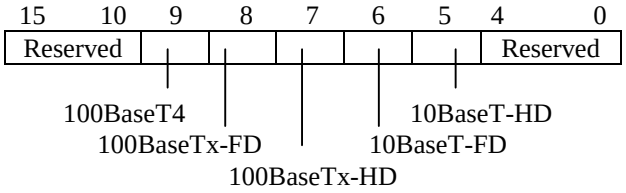
7.3.2.2.2 21140 Extended Format – Block Type 1

This block type can be used to describe MII PHY chips only.

	7	6	0	Byte Offset
BLOCK DATA	F(1)	Length		0
	Type (0x01)			1
	PHY Number			2
	GPR Sequence Length			3
	GPR Sequence (GPR Sequence Length bytes)			4
	Reset Sequence Length			4 + GPR_len
	Reset Sequence (Reset Sequence Length bytes)			5 + GPR_len
	Media Capabilities (2 bytes)			5+GPR_len + Rst_len
	Nway Advertisement (2 bytes)			7+GPR_len + Rst_len
	FDX Bit Map (2 bytes)			9+GPR_len + Rst_len
	TTM Bit Map (2 bytes)			11+GPR_len + Rst_len 12+GPR_len + Rst_len

Field	Size (Bits)	Description
Length	8	The value in this field is always 12 + GPR_Length + Reset_Length for block type 1.
Format Indicator	1	The value in this field must be 1 for Extended Format.
Type	8	This field displays the Block Type: 0x01 .
PHY Number	8	This value represents the index of the PHY chip on the board. The PHY value is determined by the chip address: the lowest address is chip #0, the next being chip #1 etc. The indexes must be contiguous, starting at 0 (that is: 0, 1, 2 etc. Sequences such as 1, 2, 3 etc. or 0, 1, 3 etc. will not function correctly). If there is an external MII connector on the board, it must be described in the last block (and is assigned the highest PHY number), despite the fact that the MII specification determines that its address must be zero.
GPR Length	8	This field indicates the number of bytes in the GPR sequence field. A GPR length of zero indicates that no value needs to be written to the General Purpose Register in order to select and activate this PHY chip. Note that it is not recommended to set GPR pins as outputs and not provide a data sequence. This may lead to unpredictable behavior because output values are unpredictable.

GPR Sequence		This is the sequence of data to be written to the General Purpose Register in order to operate this PHY chip. The sequence is a stream of bytes that is written every time the controller switches media to one supported by this PHY chip. The bytes will be written in the order displayed here (one byte at a time). The number of bytes in this field is contained in the GPR_Length field as above.
Reset Length	8	The value in this field displays the number of bytes in the Reset Sequence field. A length of zero indicates that there is no reset sequence in this block. If there is a reset sequence info block (Block Type 5), there should be no reset sequence in this block.
Reset Sequence		This is the sequence of data to be written to the General Purpose Register in order to reset this PHY chip. The bytes will be written in the order displayed here. The number of bytes in this field is contained in the Reset_length field as above. The reset sequence will be executed the first time this PHY chip is selected, before the GPR sequence is executed.
Media Capabilities	16	This field contains a bit map that describes the media being supported for this specific PHY chip. Each bit in the map represents a different medium, as shown in the following bit map:  A bit that is set indicates the medium is supported, a cleared bit indicates that it is not. This allows board designers to select and support a subset of the total capabilities initially supported by the PHY chip itself, if they so desire.

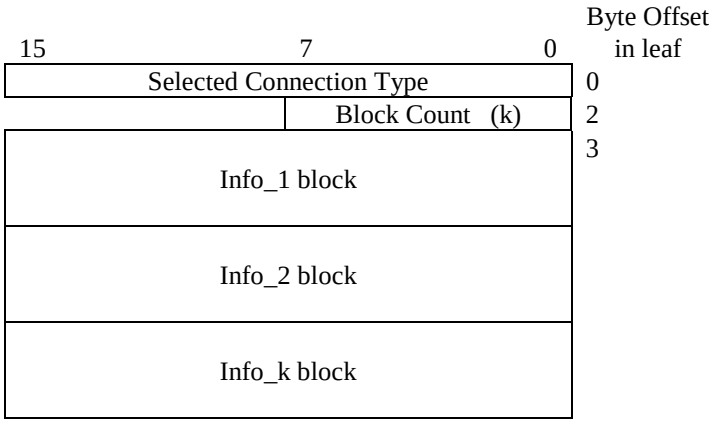
Nway Advertisement	16	This bit map is for use in AutoNegotiation advertisement. This allows board designers to determine those capabilities that they wish the PHY should advertise during the AutoNegotiation process. These may well be a subset of the supported capabilities, but should never include media that aren't supported in the capability mask above. The bit map for the Nway advertisement follows the Nway advertisement register as defined in the MII specification: 
Full_Duplex Bits	16	This field indicates the value to be written to the CSR6<FDX> for each medium (according to the bit map of media capabilities).
Tx Threshold Mode Bits	16	This field indicates the value to be written to the CSR6<TTM> for this medium (according to the bit map of media capabilities).

7.3.2.2.3 21140 Extended Format – Block Type 5

This block type provides information for resetting devices (chips) connected to the controller, via the General Purpose Register. There can be one such block per controller on the board.
It is identical to Block Type 5 used for the 21143. See Section 7.5.2.1.4 for details.

7.4 21142 Info Leaf

7.4.1 21142 Info Leaf Format



Field	Size (Bytes)	Description
Selected Connection Type	2	The connection type utilized by the controller is usually selected by the user in the drivers' configuration files. The purpose of this field is to allow information that cannot be modified by the setup utilities in the configuration files to be saved in the SROM instead. The standard case (where the media selection information is stored in the drivers' configuration files) is for this field to be set to one of the following values, depending on the board's capabilities: 0x0800 : Powerup AutoSense and Dynamic AutoSense if the board supports it. 0x8800 : Powerup AutoSense only

		For those setup utilities using the SROM, the possible values are: 0x0000 10BaseT (Twisted Pair) 0x0100 10BaseT with Nway 0x0204 10BaseT Full_Duplex 0x0001 10Base2 (BNC) 0x0002 10Base5 (AUI) 0x0003 100BaseTx (Symbol Scrambler) 0x0205 100BaseTx Full_Duplex 0x0006 100BaseT4 0x0007 100BaseFx (Fiber) 0x0208 100BaseFx Full_Duplex 0x0009 MII 10BaseT 0x020A MII 10BaseT Full_Duplex 0x000D MII 100BaseTx 0x020E MII 100BaseTx Full_Duplex 0x000F MII 100BaseT4 0x0010 MII 100BaseFx 0x0211 MII 100BaseFx Full_Duplex 0x0800 Powerup AutoSense, Dynamic AutoSense 0x8800 Powerup AutoSense only 0xFFFF No selected media information Any other value is invalid and may cause unpredictable results.
Block Count	1	This field specifies the number of Info blocks present for this controller.
Info_n block	Media Dependent	Each of these fields describes a particular supported medium/PHY chip. There is one such field for every supported non-MII medium and one for every MII PHY chip. See Section 7.4.2 for further details. The order of the info blocks defines their precedence during autosensing. The first entry is the medium/PHY chip of lowest precedence and is the one that will be checked last. The final entry in the list is the medium/PHY chip with highest precedence and is checked first. Note: The Info Block is always in extended format. Refer Section 7.3.2.2.

7.4.2 21142 Info Block Format

The 21142 supports only extended info blocks.

7.4.2.1.1 21142 Extended Format – Block Type 2

This block type describes SIA media only.

	7	6		0	Byte Offset
BLOCK DATA	F(1)	Length			0
	Type (0x2)				1
	Res	Ext	Media Code		2
	Media Specific Data (6 bytes)				3
	General Purpose Port Control (2 bytes)				3 / 9
	General Purpose Port Data (2 bytes)				5 / 11

Field	Size (Bits)	Description
Length	7	The field length is either 0x06 or 0x0C (12) for block type 2.
Format Indicator	1	Field value must be 1 for extended format.
Type	8	Block type: 0x02.
Media Code	6	This indicates to the driver that this medium is supported by the controller. Possible values are: 0x00 10BaseT (Twisted Pair) 0x01 10Base2 (BNC) 0x02 10Base5 (AUI) 0x04 10BaseT Full_Duplex
EXT	1	When set to 1, this field indicates that the lower 16 bits of CSRs 13-15 (i.e. the SIA registers) will be set via the SROM and are not using the internal default values for this media type. The SIA setting values are provided by the Media Specific Data field.
Res	1	Reserved

<table>
<tr><td>Media Specific Data</td><td>0/48</td><td>This field contains valid data only when the EXT bit is set. When the EXT bit is clear, this field contains zero bytes.

This field provides the values of CSR13, CSR14, CSR15 that should override the driver internal defaults for this media type.

<table><tr><td>15 8</td><td>7 0</td></tr><tr><td colspan="2">CSR13<15..0></td></tr><tr><td colspan="2">CSR14<15..0></td></tr><tr><td colspan="2">CSR15<15..0></td></tr></table></td></tr>
<tr><td>General Purpose Control</td><td>16</td><td>Represents the value of the General Purpose Mask Register for this medium.

This value is controller specific, and determines the character of the General Purpose port bits - for example, which bits will be inputs, which bits will be outputs, which bits will support LEDs etc.</td></tr>
<tr><td>General Purpose Port Data</td><td>16</td><td>When this media is selected, these 16 data bits are written to the General Purpose Data Register of controller_n (21142). The value of this parameter is board and controller specific. It initializes and enables the selected media’s hardware. Data is defined by the board manufacturer.</td></tr>
</table>

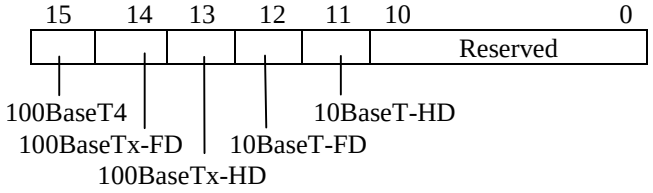
7.4.2.1.2 21142 Extended Format – Block Type 3

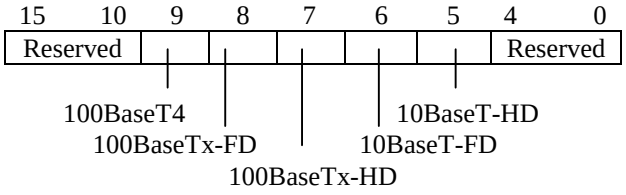
This block type describes MII PHY chips only.

BLOCK DATA

7	6	0	Byte Offset
F(1)	Length		0
Type (0x3)			1
PHY Number			2
GPR Sequence Length			3
GPR Sequence (GPR Sequence Length words)			4
Reset Sequence Length			4 + 2*GPR_len
Reset Sequence (Reset Sequence Length words)			5 + 2*GPR_len
Media Capabilities (2 bytes)			5 + 2*GPR_len + 2*Rst_len
Nway Advertisement (2 bytes)			7 + 2*GPR_len + 2*Rst_len
FDX Bit Map (2 bytes)			9 + 2*GPR_len + 2*Rst_len
TTM Bit Map (2 bytes)			11+2*GPR_len + 2*Rst_len
MII PHY Insertion/removal Indication			13+2*GPR_len + 2*Rst_len

Field	Size (Bits)	Description
Length	7	The value in this field is always 13 + 2*GPR_Length + 2*Reset_Length for block type 1.
Format Indicator	1	The value in this field must be 1 for Extended Format.
Type	8	This field displays the Block Type: 0x03
PHY Number	8	This value represents the index of the PHY chip on the board. The PHY value is determined by the chip address: the lowest address is chip #0, the next being chip #1 etc. The indexes must be contiguous, starting at 0 (that is: 0, 1, 2 etc. Sequences such as 1, 2, 3 etc. or 0, 1, 3 etc. will not function correctly). If there is an external MII connector on the board, it must be described in the last block (and is assigned the highest PHY number), despite the fact that the MII specification determines that its address must be zero.
GPR Length	8	This field indicates the number of words in the GPR sequence field. A GPR length of zero indicates that no value needs to be written to the General Purpose Register in order to select and activate this PHY chip.

GPR Sequence		This is the sequence of data to be written to the General Purpose Register in order to operate this PHY chip. The sequence is a stream of words that is written every time the controller switches media to one supported by this PHY chip. The words will be written in the order displayed here (one word at a time). The number of words in this field is contained in the GPR_Length field. The first word must always be a control word. Note that it is not recommended to set GPR pins as outputs and not provide data values. This may lead to unpredictable behavior because output values are unpredictable.
Reset Length	8	The value in this field displays the number of words in the Reset Sequence field. A length of zero indicates that the PHY chip is not reset via the General Purpose Register.
Reset Sequence		This is the sequence of data to be written to the General Purpose Register in order to reset this PHY chip. The words will be written in the order displayed here. The number of words in this field is contained in the Reset_length field. The reset sequence will be executed the first time this PHY chip is selected, before the GPR sequence is executed. The first word must always be a control word. Note that it is not recommended to set GPR pins as outputs and not provide data values. This may lead to unpredictable behavior because output values are unpredictable.
Media Capabilities	16	This field contains a bit map that describes the media being supported for this specific PHY chip. Each bit in the map represents a different medium, as shown in the following bit map representation:  A bit that is set indicates the medium is supported, a cleared bit indicates that it is not. This allows board designers to select and support a subset of the total capabilities initially supported by the PHY chip itself, if they so desire.

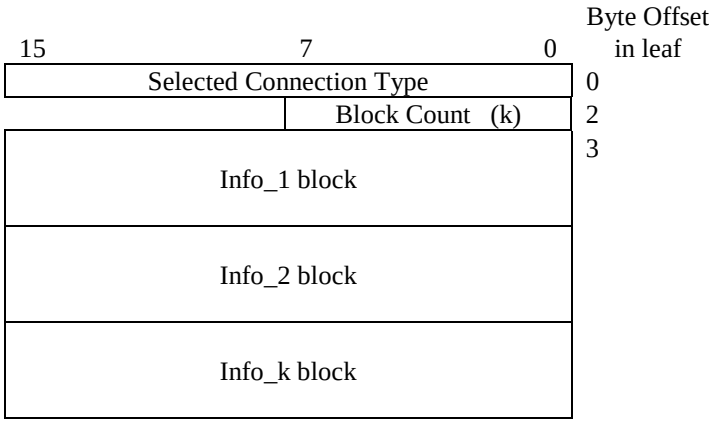
Nway Advertisement	16	This bit map is for use in AutoNegotiation advertisement. This allows board designers to determine those capabilities that they wish the PHY should advertise during the AutoNegotiation process. These may well be a subset of the supported capabilities, but should never include media that aren't supported in the capability mask above. The bit map for the Nway advertisement follows the Nway advertisement register as defined in the MII specification: 
Full_Duplex Bits	16	This field indicates the value to be written to the CSR6<FDX> for each medium (according to the bit map of media capabilities).
Tx Threshold Mode Bits	16	This field indicates the value to be written to the CSR6<TTM> for this medium (according to the bit map of media capabilities).
MII PHY Insertion/removal Indication	8	The values in this field indicate whether insertion/removal of the MII connector generates an interrupt, and if so, they also indicate which general purpose pin is generating the interrupt. The possible values are as follows: 0x00 no insertion/removal capability of external MII connector. 0x01 MII PHY insertion/removal will cause GEP0 to issue an interrupt. 0x02 MII PHY insertion/removal will cause GEP1 to issue an interrupt. 0x03 MII PHY insertion/removal will cause both GEP0 and GEP1 to issue an interrupt. 0x80 The MII PHY is removable but no interrupt is generated when inserting or removing the MII PHY. This indicates to the driver that the MII PHY is dynamically pluggable, and may need to perform presence detection of an MII device as a result ***Note***: The GEP pins should be programmed to issue such an interrupt which is described above. This is done by writing the appropriate GEP control and data words. This is true ONLY for the values 0x01, 0x02 and 0x03; NOT for the 0x80 (since 0x80 will not cause an interrupt).

7.4.2.1.3 21142 Extended Format – Block Type 5

This block type provides information for resetting devices (chips) connected to the controller, via the General Purpose Register. There can be one such block per controller on the board.
It is identical to Block Type 5 used for the 21143. See Section 7.5.2.1.4 for details.

7.5 21143 Info Leaf

7.5.1 21143 Info Leaf Format



Field	Size (Bytes)	Description
Selected Connection Type	2	The connection type utilized by the controller is usually selected by the user in the drivers' configuration files. The purpose of this field is to allow information that cannot be modified by the setup utilities in the configuration files to be saved in the SROM instead. The standard case (where the media selection information is stored in the drivers' configuration files) is for this field to be set to one of the following values, depending on the board's capabilities: 0x0800 : Powerup AutoSense and Dynamic AutoSense if the board supports it. 0x8800 : Powerup AutoSense only

		For those setup utilities using the SROM, the possible values are: 0x0000 10BaseT (Twisted Pair) 0x0100 10BaseT with Nway 0x0204 10BaseT Full_Duplex 0x0001 10Base2 (BNC) 0x0002 10Base5 (AUI) 0x0003 100BaseTx (Symbol Scrambler) 0x0205 100BaseTx Full_Duplex 0x0006 100BaseT4 0x0007 100BaseFx (Fiber) 0x0208 100BaseFx Full_Duplex 0x0009 MII 10BaseT 0x020A MII 10BaseT Full_Duplex 0x000D MII 100BaseTx 0x020E MII 100BaseTx Full_Duplex 0x000F MII 100BaseT4 0x0010 MII 100BaseFx 0x0211 MII 100BaseFx Full_Duplex 0x0800 Powerup AutoSense, Dynamic AutoSense 0x8800 Powerup AutoSense only 0xFFFF No selected media information Any other value is invalid and may cause unpredictable results.
Block Count	1	This field specifies the number of Info blocks present for this controller.
Info_n block	Media Dependent	Each of these fields describes a particular supported medium/PHY chip. There is one such field for every supported non-MII medium and one for every MII PHY chip. See Section 7.5.2 for further details. The order of the info blocks defines their precedence during autosensing. The first entry is the medium/PHY chip of lowest precedence and is the one that will be checked last. The final entry in the list is the medium/PHY chip with highest precedence and is checked first. Note: The Info Block is always in extended format. Refer Section 7.3.2.2.

7.5.2 21143 Info Block Format

The 21143 supports only extended info blocks.

7.5.2.1.1 21143 Extended Format – Block Type 2

This block is for describing SIA media and is identical to Block Type 2 used for the 21142. See Section 7.4.2.1.1 for details.

7.5.2.1.2 21143 Extended Format – Block Type 3

This block is for describing MII PHY media and is identical to Block Type 3 used for the 21142. See Section 7.4.2.1.2 for details.

7.5.2.1.3 21143 Extended Format – Block Type 4

This block is for describing SYM media only.

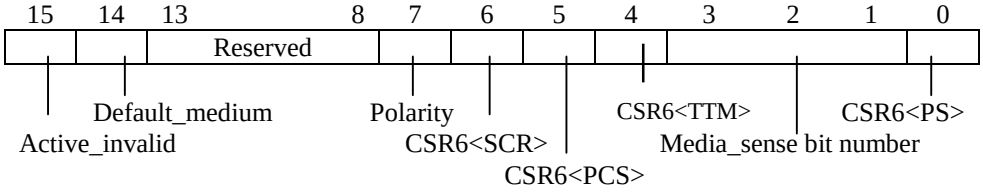
BLOCK DATA

7	6	5	0	Byte Offset
F(1)	Length			0
Type (0x4)				1
Res		Media Code		2
General Purpose Port Control (2 bytes)				3
General Purpose Port Data (2 bytes)				5
Command (2 bytes)				7

Field	Size (Bits)	Description
Length	7	The field length is 8 for block type 4.
Format Indicator	1	Field value must be 1 for extended format.
Type	8	Block type: 0x04.

Media Code	6	This field indicates to the driver that this particular SYM medium is supported by the controller. Possible values are: 0x03 100BaseTx (Symbol Scrambler) 0x05 100BaseTx Full_Duplex 0x06 100BaseT4 0x07 100BaseFx (Fiber) 0x08 100BaseFx Full_Duplex
Res	2	Reserved
General Purpose Control	16	Represents the value of the General Purpose Mask Register for this medium. This value is controller specific, and determines the character of the General Purpose port bits - for example, which bits will be inputs, which bits will be outputs, which bits will support LEDs etc.
General Purpose Port Data	16	When this media is selected, these 16 data bits are written to the General Purpose Data Register of controller_n (21143). The value of this parameter is board and controller specific. It initializes and enables the selected media's hardware. Data is defined by the board manufacturer.
Command	16	When this medium is selected, this field is used for generating the CSR6's mode bits of controller_n (21143).

A description of the command field now follows.



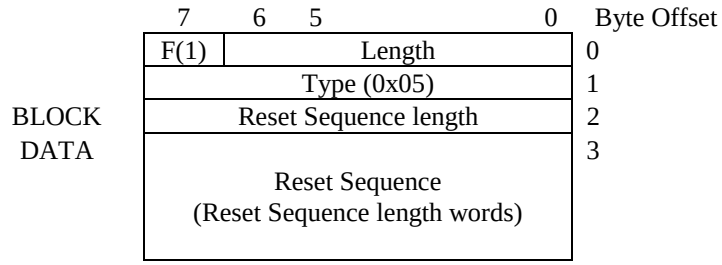
Active_invalid	If set, this bit indicates that the Media Sense bit number is not valid and that there is no media activity indication in the General Purpose Register.
Default_medium	Dynamic Autosensing will only be attempted between media for whom this bit is clear (indicating that there is a valid media sense bit to test). If set, this bit indicates the default medium that will be selected if no active link is found during the autosense process (Powerup and Dynamic). This bit is valid only if the Active_invalid bit is clear for this medium. This bit is not valid for Full_Duplex media entries. It should be set for one medium only.

Polarity	This bit indicates the polarity of the media activity indication bit in the General Purpose Register (meaningful only if the Active_invalid bit is not set). When this bit is clear, the media activity bit in the GPR is active-high i.e. it will read ’1 ’when the medium is active. When this bit is set, the media activity bit in the GPR is active-low i.e. it will read ’0 ’when the medium is active. The following table describes the activity indication:

GPR Activity Bit	Polarity Bit	Indicates the Medium is:
0	1	Not active
0	1	Active
1	0	Active
1	1	Not active

7.5.2.1.4 21143 Extended Format - Block Type 5

This block type provides information for resetting devices (chips) connected to the controller, via the General Purpose Register. There can be one such block per controller on the board.



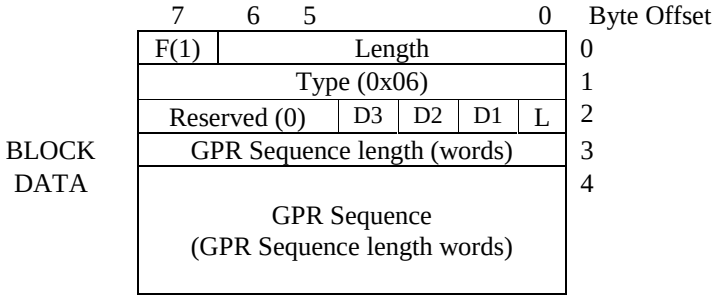
Field	Size (Bits)	Description
Length	7	The value in this field is the length of the block in bytes, not including this byte. This length should equal ("Reset Sequence length" * 2) + 2. (1 for the "Reset Sequence length" field and 1 for the "Type" field) for Block Type 5.
Format Indicator	1	The value in this field must be 1 for Extended Format.
Type	8	Block type: 0x05.
Reset Length	8	The value in this field displays the number of words in the Reset Sequence field. A length of zero indicates that no external chips are reset via the General Purpose Register.
Reset Sequence		This is the sequence of data to be written to the General Purpose Register in order to reset devices connected to the controller. The words will be written in the order displayed here. The number of words in this field is contained in the Reset_length field as above. The reset sequence will be executed at driver initialization, before any medium is selected. The first word must always be a control word. Note that it is not recommended to set GPR pins as outputs and not provide data values. This may lead to unpredictable behavior because output values are unpredictable.

7.5.2.1.5 21143 (Rev 4.1 and Above) Extended Format - Block Type 6

This block provides support for phy-shutdown upon removal of the connector, via the General Purpose Register.

More than one of the bits L, D1, D2 and D3 can be set at a time, indicating that the same GPR sequence applies to all conditions that are set.
By definition, more than one of these blocks is permitted in the SROM, provided that each of the mode bits (L, D1, D2, D3) is set at most once. However, the current drivers provided by Intel support at most one such block. The results are unpredictable if the SROM contains more than one block of type 6.

This block type applies only to devices start with 21143 Rev 4.1 and 21145.

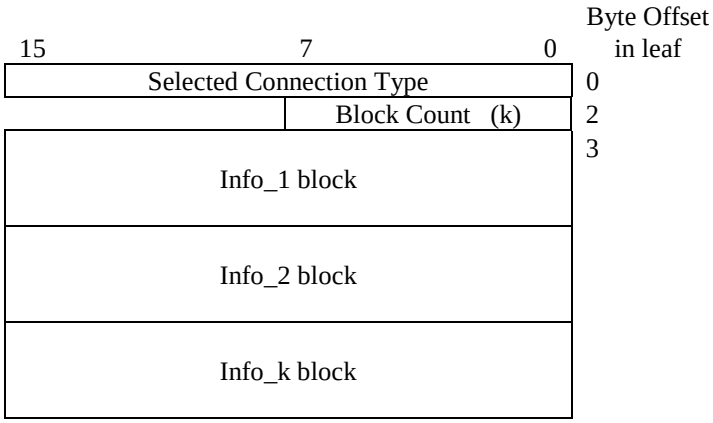


Field	Size (Bits)	Description
Length	7	The length of the block in bytes, not including this byte. This length should equal ("GPR Sequence length" * 2) + 3. (1 for the "GPR Sequence length" field, 1 for the "Type" field and 1 for the L, D1, D2, D3 and reserved byte) for Block Type 6.
Format Indicator	1	The value in this field must be 1 for Extended Format.
Type	8	Block type: 0x06.
L	1	Apply this GPR sequence when a Link Fail situation occurs. This is for use on boards with the following features: • Symbol PHY • Switch or short to identify network connector presence. • Automatic circuit for powering-up the PHY and issuing a required reset.
D1	1	Apply this GPR sequence when transitioning to the D1 power state.
D2	1	Apply this GPR sequence when transitioning to the D2 power state.

D3	1	Apply this GPR sequence when transitioning to the D3 power state.
Reserved	4	Must be zeros.
GPR Length	8	The value in this field displays the number of words in the GPR Sequence field.
GPR Sequence		This is the sequence of data to be written to the General Purpose Register in order to shut-down the Phy upon removal of the connector. The words will be written in the order displayed here. The number of words in this field is contained in the GPR_Length field as above. The first word must always be a control word. Note that it is not recommended to set GPR pins as outputs and not provide data values. This may lead to unpredictable behavior because output values are unpredictable.

7.6 21145 Info Leaf

7.6.1 21145 Info Leaf Format



Field	Size (Bytes)	Description
Selected Connection Type	2	The connection type utilized by the controller is usually selected by the user in the drivers' configuration files. The purpose of this field is to allow information that cannot be modified by the setup utilities in the configuration files to be saved in the SROM instead. The standard case (where the media selection information is stored in the drivers' configuration files) is for this field to be set to one of the following values, depending on the board's capabilities: 0x0800 : Powerup AutoSense and Dynamic AutoSense if the board supports it. 0x8800 : Powerup AutoSense only

		For those setup utilities using the SROM, the possible values are: 0x0000 10BaseT (Twisted Pair) 0x0100 10BaseT with Nway 0x0204 10BaseT Full_Duplex 0x0012 HomeRun 0x0003 100BaseTx (Symbol Scrambler) 0x0205 100BaseTx Full_Duplex 0x0006 100BaseT4 0x0007 100BaseFx (Fiber) 0x0208 100BaseFx Full_Duplex 0x0009 MII 10BaseT 0x020A MII 10BaseT Full_Duplex 0x000D MII 100BaseTx 0x020E MII 100BaseTx Full_Duplex 0x000F MII 100BaseT4 0x0010 MII 100BaseFx 0x0211 MII 100BaseFx Full_Duplex 0x0800 Powerup AutoSense, Dynamic AutoSense 0x8800 Powerup AutoSense only 0xFFFF No selected media information Any other value is invalid and may cause unpredictable results.
Block Count	1	This field specifies the number of Info blocks present for this controller.
Info_n block	Media Dependent	Each of these fields describes a particular supported medium/PHY chip. There is one such field for every supported non-MII medium and one for every MII PHY chip. See Section 7.5.2 for further details. The order of the info blocks defines their precedence during autosensing. The first entry is the medium/PHY chip of lowest precedence and is the one that will be checked last. The final entry in the list is the medium/PHY chip with highest precedence and is checked first. Note: The Info Block is always in extended format. Refer Section 7.3.2.2.

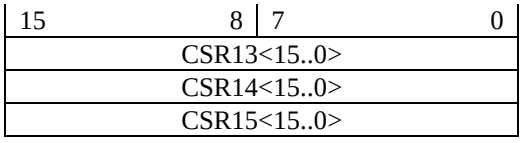
7.6.2 21145 Info Block Format

The 21145 supports only extended info blocks.

7.6.2.1.1 21145 Extended Format – Block Type 2

This block is for describing SIA media and is almost identical to Block Type 2 used for the 21142, except that in the 21145 it supports HomeRun media instead of BNC (10Base2) and AUI (10Base5).

	7	6		0	Byte Offset
BLOCK DATA	F(1)	Length			0
	Type (0x2)				1
	Res	Ext	Media Code		2
	Media Specific Data (6 bytes)				3
	General Purpose Port Control (2 bytes)				3 / 9
	General Purpose Port Data (2 bytes)				5 / 11

Field	Size (Bits)	Description
Length	7	The field length is either 0x06 or 0x0C (12) for block type 2.
Format Indicator	1	Field value must be 1 for extended format.
Type	8	Block type: 0x02.
Media Code	6	This indicates to the driver that this medium is supported by the controller. Possible values are: 0x00 10BaseT (Twisted Pair) 0x04 10BaseT Full_Duplex 0x12 HomeRun
EXT	1	When set to 1, this field indicates that the lower 16 bits of CSRs 13-15 (i.e. the SIA registers) will be set via the SROM and are not using the internal default values for this media type. The SIA setting values are provided by the Media Specific Data field.
Res	1	Reserved
Media Specific Data	0/48	This field contains valid data only when the EXT bit is set. When the EXT bit is clear, this field contains zero bytes. This field provides the values of CSR13, CSR14, and CSR15 that should override the driver internal defaults for this media type. 
General Purpose Control	16	Represents the value of the General Purpose Mask Register for this medium. This value is controller specific, and determines the character of the General Purpose port bits - for example, which bits will be inputs, which bits will be outputs, which bits will support LEDs etc.
General Purpose Port Data	16	When this media is selected, these 16 data bits are written to the General Purpose Data Register of controller_n (21145). The value of this parameter is board and controller specific. It initializes and enables the selected media's hardware. Data is defined by the board manufacturer.

7.6.2.1.2 21145 Extended Format – Block Type 3

This block is for describing MII PHY media and is identical to Block Type 3 used for the 21142. See Section 7.4.2.1.2 for details.

7.6.2.1.3 21145 Extended Format – Block Type 4

This block is for describing SYM media and is identical to Block Type 4 used for the 21143. See Section 7.5.2.1.3 for details.

7.6.2.1.4 21145 Extended Format - Block Type 5

This block is for describing Reset Sequence and is identical to Block Type 5 used for the 21143. See Section 7.5.2.1.4 for details.

7.6.2.1.5 21145 Extended Format - Block Type 6

This block is for describing GPR Sequence for external power down and is identical to Block Type 6 used for the 21143. See Section 7.5.2.1.5 for details.

7.6.2.1.6 21145 Extended Format - Block Type 7

This block is for describing HomeRun media register values, which can be different in different board designs. The basic leaf contains the HomeRunAnalogCtrl register, and 7 Homerun registers, which are the most used. If there is a need for other HomeRun registers (in the future), they can be added, changing the Length value accordingly (each additional register, the length value should be increased by 2).

Note: In case that some of these HomeRun register values will be present in the registry as defined keywords, the registry values will supersede the srom programmed ones.

7	6	5	4	0	Byte Offset
F(1)	Length				0
Type (0x7)					1
HomeRun Analog Ctrl (2 bytes)					2
HR Reg 00 Control Register (byte)					4
HR Reg 01 Control Register (byte)					5
HR Reg 10h Noise Register (byte)					6
HR Reg 12h NSE_Floor (byte)					7
HR Reg 13h NSE_Ceiling (byte)					8
HR Reg 14h NSE_Attack (byte)					9
Res			HR Reg number		10
HR Register value					11

Field	Size (Bits)	Description
Length	7	The field length value is 0x09 if the basic leaf is implemented. For each additional HomeRun register, the length should be increased by 2.
Format Indicator	1	Field value must be 1 for extended format.
Type	8	Block type: 0x07.
HomeRunAnalog Ctrl	16	Represents the value of the HomeRunAnalogCtrl register (CSR13<31..16>).
HR reg 00 Control Register	8	Represents the value of the HomeRun Control Register (lower byte) - HR00.
HR reg 01 Control Register	8	Represents the value of the HomeRun Control Register (upper byte) - HR01.
HR reg 10 Noise Register	8	Represents the value of the HomeRun Noise Register – HR10.
HR reg 12 NSE_Floor	8	Represents the value of the HomeRun NSE_Floor – HR12.
HR reg 13 NSE_Ceiling	8	Represents the value of the HomeRun NSE_Ceiling – HR13.
HR reg 14 NSE_Attack	8	Represents the value of the HomeRun NSE_Attack – HR14.
HR reg Number	5	This field is valid only if the field length value is greater than 0x0B (11). It represents the HomeRun register number which value is present in the coming byte.
Res	3	Reserved.
HR Register value	8	Represents the value of the HomeRun Register which number is contained in the *HR Reg Number* field.

Appendix A – SROM_CRC Calculation Algorithm

```
unsigned short CalcSromCrc(unsigned char *SromData);

#define  DATA_LEN        126        // 1024 bits SROM

struct {
            unsigned char  SromData[DATA_LEN];
            unsigned short SromCRC;
            } Srom;
main()
{
 Srom.SromCRC = CalcSromCrc(&Srom.SromData);
}

unsigned short CalcSromCrc(unsigned char *SromData)
{
#define POLY 0x04C11DB6L

  unsigned long crc = 0xFFFFFFFF;
  unsigned long FlippedCRC = 0;

  unsigned char CurrentByte;
  unsigned Index;
  unsigned Bit;
  unsigned Msb;
  int i;

  for (Index = 0; Index < DATA_LEN; Index++)
  {
    CurrentByte = SromData[Index];

    for (Bit = 0; Bit < 8; Bit++)
    {
      Msb = (crc >> 31) & 1;
      crc <<= 1;

      if (Msb ^ (CurrentByte & 1))
      {
        crc ^= POLY;
        crc |= 0x00000001;
      }

      CurrentByte >>= 1;
    }
  }

  for (i = 0; i < 32; i++)
  {
    FlippedCRC <<= 1;
    Bit = crc & 1;
    crc >>= 1;
    FlippedCRC += Bit;
  }
```

```
  crc = FlippedCRC ^ 0xFFFFFFFF;
  return (crc & 0xFFFF);
}
```

Appendix B – ID_BLOCK_CRC Calculation Algorithm

```
/*
** This program calculates the CRC which sums the Serial ROM ID Block header and the Magic
** Information Block.
** In the case of the ID Block header, this serial ROM header of 9 words is read upon reset of
** the chip. If the CRC result of these 9 words equals 0, it means the data has
** been read correctly.
** In the case of the Magic info block, this data is read by the chip upon entering
** Magic mode. If the CRC result of these 16 words is equal zero, it means the
** data has been read correctly and the chip can enter Magic mode.
**
** CRC is an 8 bit crc. Polynom is  x^8 + x^2 + x^1 + 1 .
** Note that contrary to a regular CRC, this CRC is calculated on the data stream
** from MSB 1'st to LSB. This is due to the nature of the SROM data stream
** which flows in this manner.
**
** Predefined SROM header:
**
**   WORD#      Meaning
**   ----------       ----------
**     0      Subsystem vendor ID
**     1      Subsystem ID
**     2      CIS pointer LOW word
**     3      CIS pointer HIGH word
**     4      reserved (value = 0)
**     5      reserved (value = 0)
**     6      reserved (value = 0)
**     7      reserved (value = 0)
**     8      High byte is reserved (value = 0), Low byte = CRC
**
** Predefined Magic block:
**
**   WORD#      Meaning
**   ----------       ----------
**     0      reserved (value = 0)
**     1      reserved (value = 0)
**     2      reserved (value = 0)
**     3      WOL IEEE addr low word
**     4      WOL IEEE addr middle word
**     5      WOL IEEE addr high word
**     6      WOL Command word
**     7      reserved (value = 0)
**     8      reserved (value = 0)
**     9      reserved (value = 0)
**    10      reserved (value = 0)
**    11      reserved (value = 0)
**    12      reserved (value = 0)
**    13      reserved (value = 0)
**    14      reserved (value = 0)
**    15      High byte is reserved (value = 0), Low byte = CRC
*/

main()
```

```
{
#define POLY 0x6

#define LEN 9                /* for ID Block */
#define LEN 16               /* for WOL block */

  unsigned short DAT[LEN];

  int i,Word,n;
  char Bit;
  unsigned char BitVal;
  unsigned char crc;

  n=0;
  crc = -1;

  for (Word=0; Word<LEN; Word++)
  for (Bit=15; Bit>=0; Bit--)
     {

          if ((Word == (LEN-1)) && (Bit == 7))
            {
             /*
             ** Insert the correct CRC result into input data stream in place.
             */
             DAT[LEN-1] = (DAT[LEN-1] & 0xff00) | (unsigned short)crc;
             break;
            }

          n++;

          BitVal = ((DAT[Word] >> Bit) & 1) ^ ((crc >> 7) & 1);
          crc = crc << 1;
          if (BitVal == 1)
            {
             crc ^= POLY;
             crc |= 0x01;
            }
     }
```

Appendix C – CIS pointer setting from SROM

CIS (Card Information Structure) is used in CardBus applications to store card's information. This information is a structure of bytes used by the system software. This specification defines two ways to store the CIS data (known as *Tuples*) in CardBus applications:

- EXPANSION ROM
- SROM

The CIS pointer register is defined in the PCI configuration space. It has a standard format defined in CardBus specification.

31 28	27 3	2 1 0
ROM Image	Address Space Offset	Address Space Indicator

CIS Pointer Layout (located in PCI configuration space)

The Address Space Indicator values can be one of two:

- 2 - CIS data is stored in SROM
- 7 - CIS data is stored in EXPANSION ROM

CIS data stored in SROM (Address Space Indicator = 2)

Use this option in case size of CIS data (Tuples) of a single function (Ethernet) or two functions (Ethernet and Modem) can fit in SROM.

To use this option, the SROM CIS Pointer Low bits 2:0 have to be set to 2 (010 binary). Tulpes are read in memory accesses mapped by CBMA base address register. The SROM can be of 1Kbits (128 bytes) or of 4Kbits (512 bytes). CBMA decodes the entire SROMs in memory range offsets of 200h:3FFh from CBMA value.

CardBus CIS Pointer Low & CardBus CIS Pointer High contain the values that are used to construct the CIS pointer register in each function's configuration space. Each function get *its own* CIS pointer value. This means that each function's CIS data is accessed at different offset in the same SROM. ROM Image number is not used if CIS points to memory space and hence this field's bits should be zeroed in the SROM.

The CIS pointer register for the Ethernet function is constructed by using the SROM CIS pointer field according to the following drawing:

31 28	27 11	10 3	2 1 0
ROM Image	Address Space Offset		Address Space Indicator
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0		SROM CIS Pointer Low <15:8>	0 1 0

CIS Pointer register (located in PCI configuration space of *Ethernet* function) layout - SROM application

Note: SROM CIS Pointer Low bits 7:3 are MBZ.

The CIS pointer register for the Modem function is constructed by using the SROM CIS pointer field according to the following drawing:

31 28	27 11	10 3	2 1 0
ROM Image	Address Space Offset		Address Space Indicator
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0		SROM CIS Pointer High <7:0>	0 1 0

CIS Pointer register (located in PCI configuration space of *Modem* function) layout - SROM application

Note: - SROM CIS Pointer High bits 15:8 are MBZ.
- If Modem function is disabled (ModemEnable = 0), then SROM CIS Pointer High bits 7:0 are MBZ.

Calculating the value of the CIS pointer field for CIS in SROM

Assume we want to locate the Ethernet CIS block at byte address 90h in the SROM and the Modem CIS block at byte address 128h (both 90h and 128h are above the 80h size of the board info).

Ethernet CIS block Offset within the SROM = 90h
SROM Offset from CBMA = 200h

Ethernet CIS block offset from CBMA = 290h

Configuration Space CIS pointer for Ethernet CIS block:
Offset = 290h = 0000 0000 0000 0000 0000 0010 1001 0000
SROM indicator: 010
The pointer: 0000 0000 0000 0000 0000 0010 1001 0010
Bits<10:3>: 010 1001 0

Modem CIS block Offset within the SROM = 128h
SROM Offset from CBMA = 200h

Modem CIS block offset from CBMA = 328h

Configuration Space CIS pointer for Modem CIS block:
Offset = 328h = 0000 0000 0000 0000 0000 0011 0010 1000
SROM indicator: 010
The pointer: 0000 0000 0000 0000 0000 0011 0010 1010
Bits<10:3>: 011 0010 1

The SROM CIS Pointer field:

0	0	6 | 5	5 | 3	0 | 2		
0000	0000	011 0010 1	010 1001 0	00000	010	= 00655202h
MBZ	MBZ	Modem CIS	Ethernet CIS	MBZ	SROM	

The SROM CIS field the indicates that the Ethernet CIS block is at offset 90h in the SROM and the Modem CIS block is at offset 128h in the SROM is: 00655202h.

CIS data stored in Expansion ROM (Address Space Indicator= 7)

Use this option in case size of CIS data (Tuples) of a single function (Ethernet) or two functions (Ethernet and Modem) is too big to fit in SROM.

To use this option, the SROM CIS Pointer Low bits 2:0 have to be set to 7 (111 binary).

CardBus CIS Pointer Low & CardBus CIS Pointer High contain the values that are used to construct the CIS pointer register in each function's configuration space. Both functions get *same* CIS pointer value. This means that both functions' CIS data is accessed at 'same' offset and 'same' ROM Image number (from host software point of view). Each function has half of the physical EXPANSION ROM size. The Ethernet function gets the first half while the Modem function gets the second half of the physical EXPANSION ROM. The Ethernet or the Modem's half EXPANSION ROM is selected by the MSB bit of the EXPANSION ROM address lines according to its defined size.

The CIS pointer register for each function is constructed by using the SROM CIS pointer field according to the following drawing:

31 28	27 16	15 3	2 1 0
ROM Image	Address Space Offset		Address Space Indicator
SROM CIS Pointer High <15:0>		SROM CIS Pointer Low <15:3>	1 1 1

CIS Pointer register (located in PCI configuration space of *each* function) layout - EXPANSION ROM application