Sundance Technology

PRELIMINARY draft 2

ST201
Fast Ethernet MAC

FEATURES

- **Single chip 10/100BASE, half or full duplex Ethernet Media Access Controller**
- **IEEE 802.3u compliant MII**
- **IEEE 802.3x full duplex flow control**
- **PCI Bus master scatter/gather DMA on any byte boundary**
- **On-chip transmit and receive FIFO buffers**
- **On-chip LED drivers**
- **Power management capabilities for ACPI 1.0 compliant systems**
- **WakeOnLAN support**
- **Management statistics gathering**
- **IP multicast receive and filter support using 64 bit hash table**
- **Receive early interrupt**
- **Transmit polling**
- **Auto pad insertion for short packets**
- **Programmable minimum Inter Packet Gap**
- **Programmable transmit and receive FIFO watermarks**
- **On-chip crystal oscillator**
- **3.3V CMOS with 5V tolerant I/O**
- **0.35μm technology**
- **128-pin PQFP**

GENERAL DESCRIPTION

The ST201 is a single-chip, full duplex, 10/100Mbps Ethernet MAC incorporating a 32-bit PCI including bus master support. The ST201 is designed for use in a variety of applications ranging from workstation NICs, networking equipment such as switches or routers, and other systems utilizing a PCI bus which require network connectivity to an Ethernet or Fast Ethernet LAN.

The ST201 includes a PCI bus interface unit, IEEE 802.3 compliant MAC, transmit and receive FIFO buffers, IEEE 802.3u compliant MII, serial Electrically EEPROM interface, expansion ROM interface, and LED drivers.

The ST201 implements a rich set of control and status registers. Accessible via the PCI interface, these registers provide a host system visibility into the features and operating state of the ST201. Network management statistics are also recorded, and host access to registers of the PHY device are facilitated through the ST201's PCI interface.

The ST201 supports several features for use in "Green PCs" or systems where control over system power consumption is desired. The ST201 supports several power down states, and the ability to issue a system "wake event" via reception of unique, user defined Ethernet frames. In addition, the ST201 can assert a wake event in response to changes in the Ethernet link status.

Publication: 2 Rev: A
Date: November 1998

BLOCK DIAGRAM

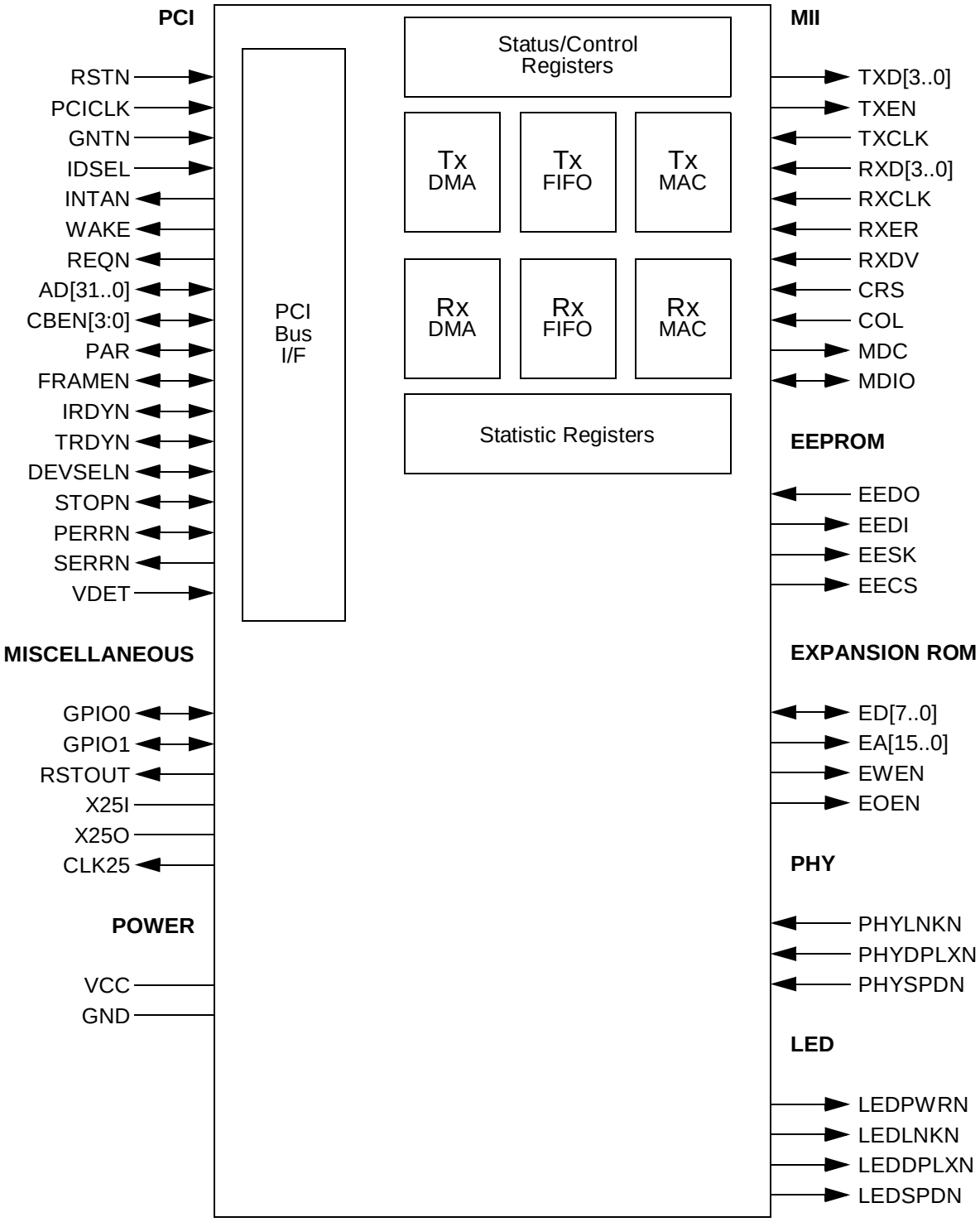
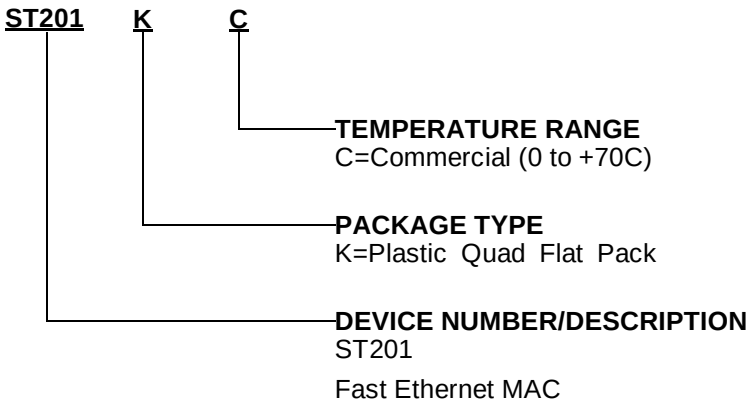


FIGURE 1: ST201 Block Diagram

ORDERING INFORMATION

Sundance products are available in several combinations of packages and operating temperature ranges. The order number is formed by a combination of the elements below



PIN DIAGRAM

PIN DESIGNATIONS

PIN NO.	PIN NAME	PIN NO.	PIN NAME	PIN NO.	PIN NAME	PIN NO.	PIN NAME
1	VCC (5V)	33	AD9	65	EA2	97	RXCLK
2	CBEN3	34	GND (5V)	66	EA3	98	RXDV
3	IDSEL	35	AD8	67	EA4	99	RXD0
4	AD23	36	CBEN0	68	EA5	100	RXD1
5	AD22	37	AD7	69	EA6	101	RXD2
6	AD21	38	AD6	70	EA7	102	RXD3
7	AD20	39	AD5	71	EA8	103	GND (5V)
8	GND (5V)	40	GND (3.3V)	72	EA9/ LEDPWRN	104	MDC
9	AD19	41	VDET	73	EA10/ LEDLNKN	105	GND (3.3V)
10	AD18	42	AD4	74	EA11/ LEDDPLXN	106	MDIO
11	AD17	43	VCC (3.3V)	75	EA12	107	PHYLNKN
12	AD16	44	AD3	76	GND (5V)	108	VCC (3.3V)
13	CBEN2	45	GND (5V)	77	EA13/EEDO	109	PHYSPDN
14	VCC (5V)	46	VCC (5V)	78	EA14/EEDI	110	VCC (5V)
15	FRAMEN	47	AD2	79	EA15/EESK	111	INTAN
16	IRDYN	48	AD1	80	EECS	112	RSTN
17	GND (5V)	49	AD0	81	X25I	113	PCICLK
18	TRDYN	50	EWEN	82	X25O	114	GNTN
19	DEVSELN	51	EOEN	83	VCC (5V)	115	REQN
20	STOPN	52	GND (3.3V)	84	RSTOUT	116	GND (3.3V)
21	PERRN	53	ED7/GPIO1	85	PHYDPLXN	117	WAKE
22	SERRN	54	ED6/GPIO0	86	CLK25	118	GND (5V)
23	PAR	55	GND (5V)	87	CRS	119	AD31
24	CBEN1	56	ED5	88	COL	120	AD30
25	GND (5V)	57	VCC (3.3V)	89	TXD3	121	AD29
26	AD15	58	ED4	90	TXD2	122	VCC (3.3V)
27	AD14	59	ED3	91	TXD1	123	AD28
28	VCC (5V)	60	ED2	92	TXD0	124	AD27
29	AD13	61	ED1	93	TXEN	125	AD26
30	AD12	62	ED0	94	GND (5V)	126	AD25
31	AD11	63	EA0	95	TXCLK	127	GND (5V)
32	AD10	64	EA1	96	RXER	128	AD24

TABLE 1: ST201 Pin Designations

PIN DESCRIPTIONS

PIN NAME	PIN TYPE	PIN DESCRIPTION
PCI INTERFACE		
RSTN	INPUT	Reset, asserted LOW. RSTN will cause the ST201 to reset all of its functional blocks. RSTN must be asserted for a minimum duration of 10 PCICLK cycles.
PCICLK	INPUT	PCI Bus Clock. This clock is used to drive the PCI bus interfaces and the internal DMA logic. All bus signals are sampled on the rising edges of PCICLK. PCICLK can operate from 0MHz to 33MHz.
GNTN	INPUT	PCI Bus Grant, asserted LOW. GNTN signals access to the PCI bus has been granted to ST201.
IDSEL	INPUT	Initialization Device Select. The IDSEL is used to select the ST201 during configuration read and write transactions.
INTAN	OUTPUT	Interrupt Request, asserted LOW. The ST201 asserts INTAN to request an interrupt, when any one of the programmed interrupt event occurs.
WAKE	OUTPUT	Wake Event, assertion level is programmable (see I/O Registers section, WakeEvent bit 3). The ST201 asserts WAKE to signal the detection of a wake event.
REQN	OUTPUT	Request, asserted LOW. The ST201 asserts REQN to request PCI bus master operation.
AD[31..0]	IN/OUT	PCI Bus Address/Data. Address and data are multiplexed on the AD pins. The AD pins carry the physical address during the first clock cycle of a transaction, and carry data during the subsequent clock cycles.
CBEN[3..0]	IN/OUT	PCI Bus Command/Byte Enable, asserted LOW. Bus command and byte enables are multiplexed on the CBEN pins. CBEN specify the bus command during the address phase transaction, and carry byte enables during the data phase.
PAR	IN/OUT	Parity. PCI Bus parity is even across AD[31..0] and CBEN[3..0]. The ST201 generates PAR during address and write data phases as a bus master, and during read data phase as a target. It checks for correct PAR during read data phase as bus master, during every address phase as a bus slave, and during write data phases as a target.
FRAMEN	IN/OUT	PCI Bus Cycle Frame, asserted LOW. FRAMEN is an indication of a transaction. It is asserted at the beginning of the address phase of the bus transaction and de-asserted before the final transfer of the data phase of the transaction.
IRDYN	IN/OUT	Initiator Ready, asserted LOW. A bus master asserts IRDYN to indicate valid data phases on AD[31..0] during write data phases, indicates it is ready to accept data during read data phases. A target will monitor IRDYN.

TABLE 2: ST201 Pin Descriptions

PIN NAME	PIN TYPE	PIN DESCRIPTION
TRDYN	IN/OUT	Target Ready, asserted LOW. A bus target asserts TRDYN to indicate valid read data phases, and to indicate it is ready to accept data during write data phases. A bus master will monitor TRDYN.
DEVSELN	IN/OUT	Device Select, asserted LOW. The ST201 asserts DEVSELN when it is selected as a target during a bus transaction. It monitors DEVSELN for any target to acknowledge a bus transaction initiated by the ST201.
STOPN	IN/OUT	Stop, asserted LOW. STOPN is driven by the slave target to inform the bus master to terminate the current transaction.
PERRN	IN/OUT	Parity Error, asserted LOW. The ST201 asserts PERRN when it checks and detects a bus parity errors. When it is generating PAR output, the ST201 monitors for any reported parity error on PERRN.
SERRN	OUTPUT	System Error, asserted LOW.
VDET	INPUT	Power Detect. The ST201 detects PCI bus power supply loss when VDET is LOW.
MII INTERFACE		
TXD[3..0]	OUTPUT	Transmit Data. This is the 4-bit transmit data, from the transmit MAC to the physical layer device. TXD[3..0] are synchronized to the TXCLK.
TXEN	OUTPUT	Transmit Enable. When asserted, TXEN indicates to the PHY that TXD[3..0] carry valid transmit data. TXEN is asserted with the first nibble of the preamble until the last nibble of the frame data. TXEN is synchronous with TXCLK.
TXCLK	INPUT	Transmit Clock. TXCLK is a continuous clock supplied by the PHY to synchronize the TXD transfer. Nominal rate of TXCLK is 25MHz for 100Mbps PHY and 2.5MHz when the PHY operates at 10Mbps.
RXD[3..0]	INPUT	Receive Data. RXD[3..0] is the receive data from the PHY. RXD[3..0] are synchronized to RXCLK.
RXCLK	INPUT	Receive Clock. RXCLK provides the timing reference for RXD, RXER, and RXDV signals. It is supplied by the PHY based on the receive clock recovery circuit. Nominal rate for RXCLK is 25MHz (for 100Mbps) and 2.5MHz (for 10Mbps).
RXER	INPUT	Receive Error. RXER is an indication from the PHY when it detects coding errors, or other types of PHY layer errors during frame data reception. RXER is synchronous with RXCLK.
RXDV	INPUT	Receive Data Valid. RXDV signals valid frame data is present on the RXD[3..0] pins. The PHY asserts RXDV before the SFD, and de-asserts it after the last data nibble of the frame. RXDV is synchronous with RXCLK.
CRS	INPUT	Carrier Sense. CRS is asserted by the PHY to signal a non-idle medium, with either transmit or receive activity detected. CRS is asynchronous to RXCLK and TXCLK.

TABLE 2: ST201 Pin Descriptions

PIN NAME	PIN TYPE	PIN DESCRIPTION
COL	INPUT	Collision. COL is asserted by the PHY to a signal collision condition is detected on the physical medium. COL is asynchronous to RXCLK and TXCLK.
MDC	OUTPUT	Management Data Clock. MDC is used to synchronize the read and write operations of MDIO.
MDIO	IN/OUT	Management Data Input/Output. MDIO carries management data for the management port read and write operations.
PHY INTERFACE		
PHYLNKN	INPUT	PHY Link Status, asserted LOW. PHYLNKN is driven by the physical layer device. It is asserted to signal a functional link (link up).
PHYDPLXN	INPUT	PHY Duplex Status, assertion level is programmable (see I/O Registers, PhyCtrl bit 4). PHYDPLXN is driven by the physical layer device. It is asserted to indicate a full duplex link, and de-asserted to indicate a half duplex link. PHYDPLXN is undefined when PHYLNKN is not asserted.
PHYSPDN	INPUT	PHY Speed Status. PHYSPDN is driven by the physical layer device. It is asserted to indicate a 100Mbps link, and de-asserted to indicate a 10Mbps link. PHYSPDN is undefined when PHYLNKN is not asserted.
EEPROM INTERFACE		
EEDO	INPUT	EEPROM Data Output. This pin is connected directly to the data output of the EEPROM device. (This pin is shared with EA13)
EEDI	OUTPUT	EEPROM Data In. This pin is connected directly to the data input of the EEPROM device. (This pin is shared with EA14)
EESK	OUTPUT	EEPROM Serial Clock. EESK is the clock used to synchronize the EEPROM data access with EEDI and EEDO. It is connected directly to the clock input of the EEPROM device. (This pin is shared with EA15)
EECS	OUTPUT	EEPROM Chip Select. EECS is asserted by the ST201 to access the EEPROM. It is connected directly to the chip select input of the EEPROM device.
EXPANSION ROM INTERFACE		
ED[7..0]	IN/OUT	Expansion ROM Data. The ED[7..0] provide data access to the expansion ROM.
EA[15..0]	OUTPUT	Expansion ROM Address. The EA[15..0] carry the address to the expansion ROM.
EWEN	OUTPUT	Expansion ROM Write Enable.
EOEN	OUTPUT	Expansion ROM Output Enable.
LED DRIVERS		

TABLE 2: ST201 Pin Descriptions

PIN NAME	PIN TYPE	PIN DESCRIPTION
LEDPWRN	OUTPUT	Power Status LED. (This pin is shared with EA9). The operation of this pin varies based on the setting in the I/O Registers, AsicCtrl bit 14 (the LEDMode bit). In Mode 0, LOW when power is applied, and toggling when frame transmission is in progress. In Mode 1, this pin is always LOW when power is applied.
LEDLNKN	OUTPUT	Link Status LED. (This pin is shared with EA10). The operation of this pin varies based on the setting in the I/O Registers, AsicCtrl bit 14 (the LEDMode bit). In Mode 0, LOW when a valid link exists, toggling when frame reception is in progress. In Mode 1, LOW when a valid link exists, and toggling when either frame transmission or reception is in progress.
LEDDPLXN	OUTPUT	Duplex Status LED. (This pin is shared with EA11). This pin operates independently of the I/O Registers, AsicCtrl bit 14 (the LEDMode bit). This pin is LOW when the PHY is in full duplex mode, and toggles when collisions are detected.
LEDSPDN	OUTPUT	Speed Status LED. (This pin is shared with EA12). This pin operates independently of the I/O Registers, AsicCtrl bit 14 (the LEDMode bit). This pin is LOW when the link speed is 100Mbps, and HIGH when the link speed is 10Mbps.
MISCELLANEOUS		
GPIO0	IN/OUT	General Purpose Input/Output. (This pin is shared with ED6)
GPIO1	IN/OUT	General Purpose Input/Output. (This pin is shared with ED7)
RSTOUT	OUTPUT	Reset Output, assertion level is programmable (see I/O Registers, AsicCtrl bit 15). The ST201 will assert RSTOUT when it is being reset. RSTOUT is intended to be used to reset other circuitry on the adapter.
X25I	OSCIN	25MHz Crystal Oscillator Input. The external 25MHz crystal and capacitor is connected to the on-chip crystal oscillator circuit through X25I input. Alternately, X25I can be driven by an external clock source.
X25O	OSCOUT	25MHz Crystal Oscillator Output. The external crystal and capacitor is also connected to the output of the on-chip crystal oscillator circuit through X25O. When X25I is driven by an external clock source, X25O should be left unconnected.
CLK25	OUTPUT	25MHz Clock Output. CLK25 carries the reference clock generated by the on-chip crystal oscillator. This is a free-running and continuous clock signal.
POWER AND GROUND		
VCC (5V)	POWER	+5 volts power supply.
GND (5V)	GROUND	+5 volts power return.
VCC (3.3V)	POWER	+3.3 volts power supply.
GND (3.3V)	GROUND	+3.3 volts power return.

TABLE 2: ST201 Pin Descriptions

ACRONYMS AND GLOSSARY

LAN	Local Area Network
MAC	Media Access Control Layer, or a device implementing the functions of this layer (a Media Access Controller)
PCI	Peripheral Component Interface
NIC	Network Interface Cards
FIFO	First In First Out
MII	Media Independent Interface
EPROM	Erasable Programmable Read Only Memory
EEPROM	Electrically Erasable Programmable Read Only Memory
LED	Light Emitting Diode
PHY	Physical Layer, or device implementing functions of the Physical Layer
CSMA/CD	Carrier Sense Multiple Access with Collision Detect
FCS	Frame Check Sequence
SFD	Start of Frame Delimiter
CRC	Cyclic-Redundancy-Check
IP	Internet Protocol
TFD	Transmit Frame Descriptor
RFD	Receive Frame Descriptor
DMA	Direct Memory Access
ACPI	Advanced Configuration and Power Management

STANDARDS COMPLIANCE

The ST201 implements functionality compliant with the following standards:

- IEEE 802.3u Fast Ethernet
- IEEE 802.3x Full Duplex Flow Control
- PCI Local Bus Revision 2.1
- ACPI Revision 1.0

FUNCTIONAL DESCRIPTION

The ST201 is composed of various functional blocks as shown in Figure 1. An overview of the functions performed by each block follows:

MEDIA ACCESS CONTROL

The MAC block implements the IEEE Ethernet 802.3u Media Access Control functions with 802.3x Full Duplex and Flow Control enhancements. In half duplex mode, the MAC implements the CSMA/CD. Full duplex mode by definition does not utilize CSMA/CD, allowing data to be transmitted on demand. An optional flow control mechanism in full duplex mode is provided via the 802.3x MAC Control PAUSE function. Additionally, the MAC also performs the following functions in either half or full duplex mode:

- Optional transmit FCS generation
- Padding to the minimum legal frame size
- Preamble and SFD generation
- Preamble and SFD removal
- Receive frame FCS checking and optional FCS stripping
- Receive frame destination address matching
- Support for multicast and broadcast frame reception or rejection (via filtering)
- Selective InterFrame Gap to avoid capture effect
- MAC Loopback

The MAC is responsible for generation of hardware signals to update the internal statistics counters.

MEDIA INDEPENDENT INTERFACE

The ST201 can support a variety of physical signaling schemes via the IEEE 802.3u defined MII. Through the MII, the ST201 supports Fast Ethernet (such as 100BASE-TX) as well as the legacy 10BASE-T standard. The MII provides a general-purpose interface between an 802.3u MAC and various physical layer devices, and is comprised of two independent components. The data interface provides separate, 4-bit wide paths for receive and transmit data, as well as independent clock and control signals. The management interface is a bidirectional, serial link that provides the ST201 access to registers residing within the physical layer device. The host system controls the MII management interface through the PhyCtrl register.

Since the MII is independent of the signaling method (100BASE-TX, 10BASE-T, etc.), it is possible to use it to support numerous Ethernet or Fast Ethernet LAN types depending upon the availability of MII-compliant PHY devices. The most widely available PHY devices support both 10BASE-T and 100BASE-TX through a single MII.

It is most likely that a physical layer device connected to ST201's MII will include implementation of the 802.3u Auto-Negotiation function. For instance, a PHY device may be able to auto-negotiate between 10BASE-T and 100BASE-TX. A host system attempting to determine link status should check the Auto-Negotiation function contained in the MII-based PHY device through the MII management interface of the ST201.

PCI BUS INTERFACE

The PCI Bus Interface (PBI) implements the procedures and algorithms needed to link the ST201 to a PCI bus. The ST201 can be either a PCI bus master or slave. The PBI is also responsible for managing the DMA interfaces and the host processors access to the ST201 registers. Arbitration logic within the PBI block accepts bus requests from the TxDMA Logic and RxDMA Logic. The arbiter services the four requests in the fixed priority order of:

1. RxDMA Urgent Request
2. TxDMA Urgent Request
3. RxDMA Request
4. TxDMA Request

The PBI also manages interrupt generation for a host processor.

TXDMA LOGIC

The ST201 supports a multi-frame, multi-fragment DMA gather process. Descriptors representing frames are built and linked in system memory by a host processor. The TxDMA Logic is responsible for transferring the multi-fragment frame data from the host memory into the TxFIFO.

The TxDMA Logic monitors the amount of free space in the TxFIFO, and uses this value to decide when to request a TxDMA. A TxDMABurstThresh register is used to delay the bus request until there is enough free space in the TxFIFO for a long burst.

To prevent a TxFIFO under run condition the TxDMA logic forwards an urgent request to the DMA arbiter, regardless of the TxDMABurstThresh constraint, when the number of occupied bytes in the TxFIFO drops below the value in TxDMAUrgentThresh register.

TXFIFO

The ST201 uses 2K bytes of transmit data buffer between the TxDMA Logic and Transmit MAC. When the TxDMA logic determines there is enough space available in the TxFIFO, the TxDMA Logic will move any pending frame data into the TxFIFO. The TxReleaseThresh register value determines the amount of data which must be transmitted out of the TxFIFO before the FIFO memory space occupied by that data can be released for use by another frame.

A TxReleaseError occurs when a frame experiences a collision after the TxFIFO release threshold has been crossed. The ST201 will not be able to retransmit this frame from the TxFIFO and the complete frame must be transferred from the host system memory to the TxFIFO again by TxDMA Logic.

RXDMA LOGIC

The ST201 supports a multi-frame, multi-fragment DMA scatter process. Descriptors representing frames are built and linked in system memory by a host processor. The RxDMA Logic is responsible for transferring the multi-fragment frame data from the RxFIFO to the host memory.

The RxDMA Logic monitors the number of bytes in the RxFIFO. After a number of bytes have been received, the frame is “visible”. A frame is visible if:

- The frame being received is determined not to be a runt, AND
- The number of frame bytes received has exceeded the value in the RxEarlyThresh register (if enabled), OR
- The entire frame has been received

After a frame becomes visible, the RxDMA Logic will issue a request to the DMA arbiter when the number of bytes in the RxFIFO is greater than the value in the RxDMABurstThresh. To prevent receive overruns, a RxDMA Urgent Request is made when the amount of free space in the RxFIFO falls below the value in RxDMAUrgentThresh.

RXFIFO

The ST201 uses 2K bytes of receive data buffer between the Receive MAC and RxDMA Logic. When the RxDMA Logic determines the number of bytes in the RxFIFO is greater than the value in RxEarlyThresh register, the ST201 will generate a RxEarly interrupt (if enabled) to the host. The values in RxEarlyThresh and RxDMABurstThresh also determine how many bytes of a frame must be received into RxFIFO before RxDMA Logic is allowed to begin data transfer.

When RxEarlyThresh is set to a value that is greater than the length of the received frame, a RxComplete interrupt will occur at the completion of frame reception rather than a RxEarly interrupt.

EEPROM INTERFACE

The external serial EEPROM is used for non-volatile storage of such information as the node address, system ID, and default configuration settings. As part of initialization after system reset, the ST201 reads certain locations from the EEPROM and places the data into host-accessible registers.

EXPANSION ROM INTERFACE

The ST201 provides support for an optional Expansion ROM. The ST201 supports the Atmel AT29C512 (64K x 8) Flash EPROM device.

The Expansion ROM is configured through the PCI configuration register, which maps the ROM into the memory space of the host system. The ROM contents can be scanned, copied to system RAM, and executed at system initialization time.

The ROM is also byte-read and byte-write accessible to the host CPU using the ExpRomData and ExpRomAddr registers. This allows a diagnostic program to read or modify the ROM contents without having to write to configuration registers.

OPERATION

Proper operation of the ST201 in a system requires an understanding of initialization tasks, register programming, transmit and receive behavior, interrupt handling, statistic gathering, PCI bus transactions, and power management capabilities.

INITIALIZATION

The ST201 provides several resets. The assertion of the hardware reset signal on the PCI bus causes a complete reset of the ST201. The ST201 configurations previously set are lost after reset. A similar reset is available via software using the GlobalReset bit of the AsicCtrl register. The AsicCtrl register also allows for selective reset of particular functional blocks of the ST201. See the Registers and Data Structures section for details on using the AsicCtrl register for resetting the ST201.

The external serial EEPROM is used for non-volatile storage of configuration information across ST201 resets. Shortly after reset, the ST201 will read the contents of an external EEPROM, placing the data read into the following registers:

- ConfigParm
- AsicCtrl (least significant 16 bits)
- SubsystemVendorId
- SubsystemId
- StationAddress

There are several other registers which must be configured during initialization. These registers include the ST201 PCI configuration registers which are set during a Power On Self Test (POST) routine performed by the host system. Specifically, the registers set during this stage of initialization are:

- ConfigCommand enables adapter operation by allowing it to respond to and generate PCI bus cycles. ConfigCommand is also used to enable parity error generation.
- IoBaseAddress sets the I/O base address for the ST201 registers.
- MemBaseAddress sets the memory base address for the ST201 registers.
- ExpRomBaseAddress sets the base address and size for an installed expansion ROM, if any.
- CacheLineSize indicates the system's cache line size. This value is used by the ST201 to optimize bus master data transfers.
- LatencyTimer sets the length of time the ST201 can hold the PCI bus as a bus master.
- InterruptLine maps ST201's interrupt request to a specific interrupt line (level) on the system board.
- AsicCtrl is used to setup internal operations and parameters.

The ST201 can be accessed across the PCI bus without setting the PCI registers or reading data from an external EEPROM. In this Forced Configuration mode (useful for embedded applications without an EEPROM), the ST201 is forced to the following configuration:

- I/O base address 200h
- I/O target cycles enabled
- Memory target cycles disabled
- Bus master cycles enabled
- Expansion ROM cycles disabled

REGISTER PROGRAMMING

After initializing the ST201 to facilitate communication with the host system, an additional set of registers specific to operation of the Ethernet network must be set.

The first setting relates to the Auto-Negotiation features present in most Ethernet PHY devices. Since the ST201 does not participate in the Auto-Negotiation process, the host system must communicate with the PHY device (across the MII Management Interface) to determine the link status. Once the result of Auto-Negotiation is determined, if a full duplex mode has been chosen, the host system must set the FullDuplexEnable bit in the MACCtrl register. Other modes chosen during Auto-Negotiation do not require any ST201 register settings.

The ReceiveMode register determines which types of frames, based on address matching mechanism, the ST201 will receive. The host system must program the adapter's node address into the StationAddress registers. This node address can be obtained either from the EEPROM, or the host system can set the value directly. Then, by setting the ReceiveUnicast bit in the ReceiveMode register, the ST201 will receive unicast frames whose destination address matches the value in the StationAd-

dress register. Setting the ReceiveBroadcast and ReceiveMulticast bits in the ReceiveMode register will allow the ST201 to receive all broadcast and multicast frames, respectively. The ReceiveMulticastHash bit in ReceiveMode enables a filtering mechanism for Ethernet multicast frames. This filtering mechanism uses a 64-bit hash table (HashTable register) for selective reception of Ethernet multicast frames. A CRC algorithm is applied to the destination address of incoming Ethernet multicast frames. The least significant 6 bits of the CRC result are used as an index into the hash table. The Ethernet multicast frame will be accepted by the ST201 when the corresponding hash table bit is set, otherwise the frame will be discarded. The host system must configure the hash table before any multicast frames are received using the filtering mechanism.

Additionally, Ethernet frames containing IP multicast destination addresses can also be received by setting the ReceiveIPMulticast bit in the ReceiveMode register. IP multicast, or Host Extension for IP Multicasting, datagrams map to frames with Ethernet destination addresses of 0x01005e****** (where * represents any hexadecimal value).

The MACCtrl register is used to configure parameters including full duplex, flow control, and statistics gathering.

The ST201 can operate in either half duplex or full duplex mode. In half duplex mode, the ST201 implements the CSMA/CD algorithm. CSMA/CD requires that only one node transmit at a time. If multiple nodes attempt to transmit simultaneously (indicated when both the receive and transmit signals are active) a collision will occur resulting in frame loss, and typically requiring re-transmission. In full duplex mode, the ST201 can transmit and receive frames simultaneously without incurring collisions. To configure the ST201 for full duplex mode operation, the host system must detect a full duplex physical link via the appropriate PHY device register, and must set the FullDuplexEnable bit in the MACCtrl register.

The IEEE 802.3x Full Duplex standard defines a special frame known as the PAUSE MAC Control frame. The PAUSE frame is used to implement flow control in full duplex networks allowing stations on opposite ends of a full duplex link the ability to inhibit transmission of MAC data frames for a specified period of time. The PAUSE frame format is defined as shown in Figure 1.

FIELD		LENGTH (BYTES)
DA	0x0180C2000001	6
SA		6
TYPE	0x8808	2
OPCODE	0x0001	2
PAUSE TIME		2
PAD		42

FIGURE 1: PAUSE Frame

In Figure 1, bytes within fields are transmitted left to right, bits within bytes transmitted least-significant bit first.

Whenever the FlowControlEnable bit in the MACCtrl register is set, the ST201 compares the destination address with 0x0180C2000001 and the type field with 0x8808 in all incoming frames. If both fields match, the ST201 further checks for the PAUSE opcode (0x0001) in the MAC Control Opcode field. If found, the ST201 inhibits transmission of all data frames for the time specified in the two-byte pause_time field. The pause_time field is specified in slot times relative to the current data rate; one slot time is 51.2 us at 10 Mbps, and 5.12 us at 100 Mbps. The transmission of PAUSE frames is the responsibility of the host. The MAC Control frame must be constructed by the host and placed into the TxFIFO. The driver should program the ReceiveMode register to receive PAUSE (MAC Control frame) when flow control is enabled. For end station applications, host system should only accept PAUSE frames, and not generate them. Flow control is designed to originate from network devices such as switches.

TXDMA AND FRAME TRANSMISSION

The TxDMA block transfers frame data from a host system to the ST201 based on a linked list of frame descriptors called TFDs. The frame to be transmitted is divided into data fragments (or buffers) within the host system's memory. The host system creates a list of TFDs, also in system memory, where each TFD contains the memory locations of one or more fragments of a frame as shown in Figure 2.

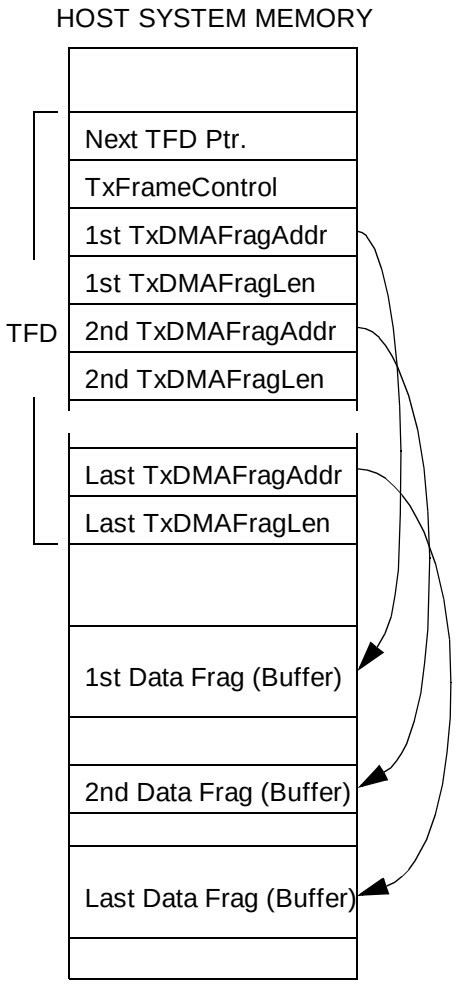


FIGURE 2: TxDMA Data Structure

The TFD format is covered in the Registers and Data Structures section.

The resulting linked list of TFDs is referred to as the TxDMAList, as shown in Figure 3.

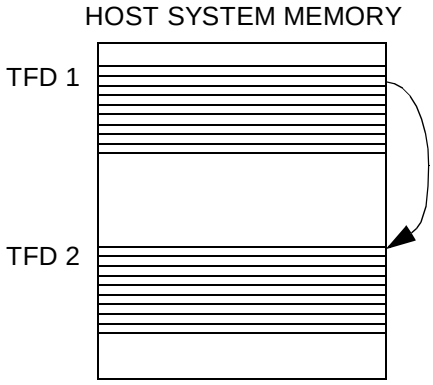


FIGURE 3: TxDMA List

After reset, the ST201 MAC transmission is disabled, until the TxEnable bit is set. Any data transferred by TxDMA Logic into the TxFIFO will stay in the TxFIFO waiting for the TxEnable bit to be set enabling transmission. After enabling data transmission, the TxDMA Logic is in the idle state. In the simple case of a single frame, the host system must create a TFD within the host system memory. This TFD must contain the addresses and lengths of the fragments containing the data to be transmitted. The host system must write zero into TxDMANextPtr since this is the only frame. The host starts the TxDMA Logic by writing the memory location (a non-zero address) of the TFD into TxDMAListPtr register. The TxDMA Logic, if is not in the TxDMAHalt state, begins transferring data into the ST201 when:

- The TxFIFO has 16 or more bytes of free space, AND
- The fragment length is less than or equal to the amount of free space in the TxFIFO, OR
- The amount of free space in the TxFIFO is greater than or equal to the value (in bytes) of 32 * TxDMABurstThresh register

If these conditions hold, the TxDMA Logic fetches the fragment addresses and fragment lengths from the TFD and writes them one at a time into the ST201 registers, which are used to control the data transfer operations. If the TxDMA Logic transfers more data than can fit into the TxFIFO, an overrun will occur.

If the host system disables data transmission (by resetting the TxDisable bit) while a frame transmission is in progress, the current frame transmission will complete before data transmission is disabled.

The TxDMAListPtr I/O register within the ST201 contains the physical address that points to the head of the TxDMAList. TxDMAListPtr must point to addresses which are on 8-byte boundaries. A value of zero in the TxDMAListPtr register implies there are no pending TFD's for the ST201 to process.

Generally, it is desirable for the host system to queue multiple frames. Multiple TFD's are linked together in a list by pointing the TxDMANextPtr of each TFD at the next TFD. The last TFD in the linked list should have a value of zero for it's TxDMANextPtr.

It is required that the host halt the TxDMA Logic by setting the TxDMAHalt bit before modifying TxDMAList or writing a new value to TxDMAListPtr (unless it is already zero). When the host has finished manipulating the list, it sets the TxDMAResume bit.

The TxDMA process returns to the idle state upon detection of a zero value for TxDMANextPtr. When a new frame is available to transfer, the host system must write the address of the new TFD into the TxDMANextPtr memory location of the last TFD, and either set the TxEnable bit, or utilize the ST201's automatic polling capability. Using automatic polling, the ST201 will monitor the TxDMANextPtr memory location until a non-zero value is found at that location in system memory. The TxDMAPollPeriod register controls this polling function, which is enabled when TxDMAPollPeriod contains a non-zero value. The value written to TxDMAPollPeriod determines the TxDMANextPtr polling interval.

The ST201 can be configured to generate TxDMAComplete interrupts on a per frame basis by setting the TxDMAIndicate bit within the TFC field of each TFD. In response to a TxDMAComplete interrupt, when data transfer by TxDMA is finished, the host system acknowledges the interrupt and returns the frame data buffers to the system. In the case of a multi-frame TxDMAList, multiple frames may have been transferred by TxDMA when the host system enters its interrupt service routine. The host system can traverse the list of TFD's, examining the TxDMAComplete bit in each TFD to determine which frames have been transferred by TxDMA.

The ST201 fetches the TFC before frame data transfer, and again at the end of TxDMA operation to examine the TxDMAIndicate bit. This allows the host system to change TxDMAIndicate while data transfer of the frame is in progress. For instance, a frame's TFD might be at the end of the TxDMAList when it starts TxDMA, so the host system would probably set TxDMAIndicate to generate an interrupt. However, if during the TxDMA process of this frame, the host system added a new TFD to the end of the list, it might clear TxDMAIndicate in the currently active TFD so that the interrupt is delayed until the next TFD.

The ST201 has the ability to automatically round up the length of a transmit frame. This is useful in some NOS environments in which frame lengths need to be an even number of words. Frame length word-alignment is performed based upon the sum of the fragment lengths specified in a TFD, and the 2-bit WordAlign field in the TFD's TFC field. Word-alignment occurs when the frame length implied by the sum of the fragment lengths is not an even multiple of the value indicated by WordAlign. The frame length is rounded up to either a word or dword boundary, depending upon the value of WordAlign. Host systems may disable frame length word-alignment by setting the WordAlign bits in the TransmitFrameControl to x1.

The MAC will initiate frame transmission (if transmission is enabled) as soon as either the entire frame is resident in the TxFIFO, or the number of bytes present in the TxFIFO is greater than the value in the TxStartThresh register.

As a frame transmits out of the TxFIFO, it is desirable to be able to release the FIFO space so that it may be used for another frame. The value programmed into TxReleaseThresh determines how much of a frame must be transmitted before its FIFO space can be released.

A TxReleaseError occurs when a frame experiences a collision after the frame's release threshold has been crossed. The ST201 will be unable to retransmit the frame and the host needs to re-start the TxDMA transfer. When a transmit error occurs, a TxComplete interrupt is generated, and the specific error is indicated by status bits in TxStatus. To recover from a transmit error, the host system must re-enable the transmitter and, in the case of an under run error, reset the transmit logic with TxReset, before subsequent transmissions can occur. When MaxCollisions or TxStatusOverflow errors occur, any pending frames in the TxFIFO are preserved (except the frame that experienced MaxCollisions). The frames are held in the TxFIFO due to the fact that MaxCollisions and TxStatusOverflow errors do not require assertion of TxReset. The preserved frames will be transmitted following a transmitter re-enable.

Since TxDMA writes into one end of the TxFIFO and the transmit MAC reads from the other end, TxDMA and transmit completes (including errors)

are independent of each other in general. A special case is when a transmit under run occurs. In this case the current frame being transmitted is the only frame in the TxFIFO. When a transmit under run occurs, the ST201 stops TxDMA operation and generates an interrupt with a TxUnderrun error flagged in TxStatus. The host system can determine which is the under run error frame by examining the current value of TxDMAListPtr. The host system can assume that all frames in the TxDMAList ahead of the under run error frame have been transmitted successfully. To recover from an under run, the host system should halt the TxDMA Logic by setting the TxDMAHalt bit, wait until TxDMAInProg and TxInProg are cleared, then issue a TxReset to reset the under run (TxFIFO and Transmit MAC). Transmission needs to be enabled (by TxEnable) again and all transmit-related thresholds (TxStartThresh in particular) should be restored. To re-transmit the frame, the host system writes the value of the under run frame's TFD into the TxDMAListPtr register.

FRAME RECEPTION AND RXDMA

The frame RxDMA mechanism is similar to the TxDMA mechanism. RxDMA is structured around a linked list of frame descriptors, called RFDs. RFDs contain pointers to the fragment buffers into which the ST201 is to place receive data, as shown in Figure 4.

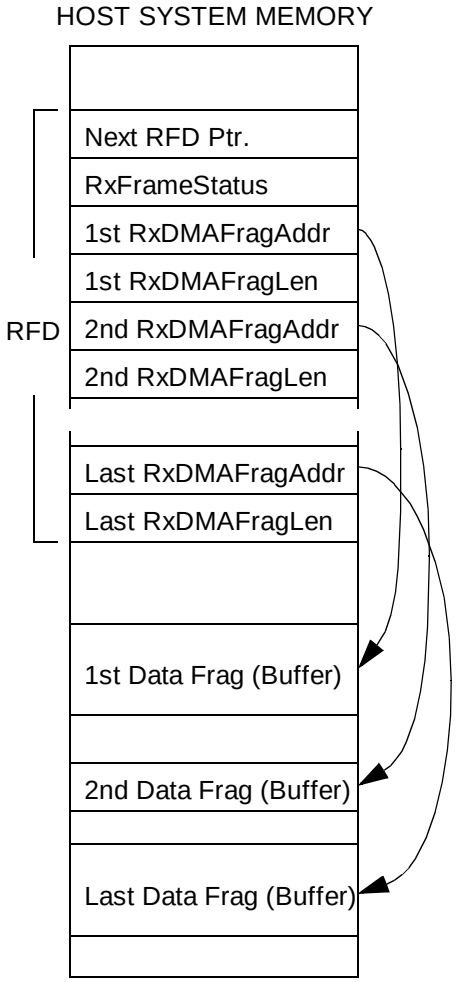


FIGURE 4: RxDMA Data Structure

The RFD format is covered in the Registers and Data Structures section.

Similar to TFDs, the resulting linked list of RFDs is referred to as the RxDMAList. One option available to RxDMA that differs from TxDMA is that the RxDMAList can be formed into a ring as shown in Figure 5. A host system can allocate a number of full size frame buffers, create a RFD for each one, and link the RFDs into a circular list. As frames are

received and transferred by RxDMA, a RxDMA-Complete interrupt will be generated for each frame.

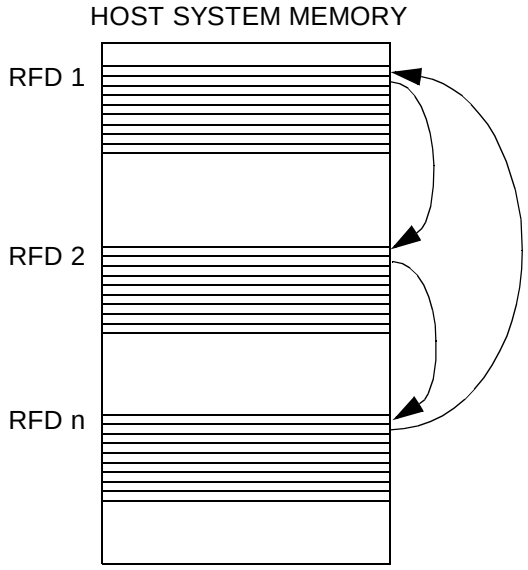


FIGURE 5: RxDMA List Shown in Ring Configuration

The host system must create a RxDMAList and the associated buffers prior to reception of a frame. One approach calls for the host system to allocate a block of full size (i.e. large enough to hold a maximum size Ethernet frame of 1518 bytes) frame buffers in system data space and create RFDs that point to them. Another approach is for the host system to request the buffers from the protocol ahead of time.

After reset, the ST201 receive function is disabled. Once the RxEnable bit is set, frames will be received according to the matching mode programmed in ReceiveMode register. Reception can be disabled by setting the RxDisable bit. If set while a frame is being received, RxDisable only takes effect after the active frame reception is finished. The receive function begins with the RxDMA Logic in the idle state. The RxDMA Logic will begin processing a RxDMAList as soon as a non-zero address is written into the RxDMAListPtr register. The host system creates a RFD with the addresses and lengths of the buffers to be used and programs the RxDMAListPtr register to point to the head of the list. The host system must program a zero into the RxDMANextPtr of the last RFD to indicate the end of the RxDMAList. When a frame is received in the RxFIFO, the ST201 fetches the fragment address and fragment length values one by one from the current RFD, and writes these values into internal registers which control data transfer operations. Similar to TxDMA, the RxDMA Logic can be controlled by the RxDMAHalt and RxDMAResume bits. The host system should set the RxDMAHalt bit before modifying the list pointers in the RxDMAList. The RxDMA Logic will return to the idle state when the RxDMAListPtr register is zero.

For RxDMA lists configured as a ring, the host system should clear the RxDMAComplete bit within the ReceiveFrameStatus field of the RFD from which the host system has finished reading data. If RxDMAPollPeriod is zero the host system should also issue a RxDMAResume in case the ST201 has halted due to detection of a set RxDMAComplete bit within the ReceiveFrameStatus field of the next RFD in the ring. If the ST201 fetches a RxDMAListPtr for a RFD that has already been used (a RFD in which the RxDMAComplete bit is set in ReceiveFrameStatus), the RxDMA Logic will either assert an implicit RxDMAHalt or, if the RxDMAPollPeriod register is set to a non-zero value, the RxDMA Logic will automatically recheck RxDMAComplete periodically until it is cleared.

The operation for adding RFDs into the RxDMAList starts with halting the RxDMA Logic by setting the RxDMAHalt bit within the DMACtrl register. The host system then updates RxDMANextPtr in the last RFD in the RxDMAList to point at the new RFD. The host system will also need to read the RxDMAListPtr, and if it was zero, write the address of the just added RFD into RxDMAListPtr and set the RxDMAResume bit within the DMACtrl register to re-start the RxDMA Logic.

The ST201 can be configured to generate a RxDMAComplete interrupt when RxDMA completes a frame reception. In response to a RxDMAComplete interrupt, the host system must examine the ReceiveFrameStatus field in the RFD of the received frame to determine the size of the frame and whether there were any errors. The host system must then copy the frame out of the receive buffers, if needed.

In general, when the host system enters its interrupt service routine, multiple frames may have been transferred by RxDMA. The host system can read RxDMAListPtr to determine which RFDs in the list have been used. The host system begins at the head of the RFD list, and traverses the list until it reaches the RFD whose address matches RxDMAListPtr. However, since I/O operations are costly, it is more efficient to use the RxDMAComplete bit in each RFD to determine which frames have been transferred by RxDMA.

Systems using the ST201 can be programmed to generate an interrupt based upon the number of bytes that have been received in a frame. The RxEarlyThresh register sets the value for early receive threshold. As soon as the number of bytes that have been received is greater than the value in RxEarlyThresh register, the ST201 will generate a RxEarly interrupt, if it is enabled, to the host. The RxEarly interrupt will only occur when the frame being received is the top frame, i.e., can be transferred by the host during reception. The RxEarlyThresh mechanism will cause one RxEarly interrupt per frame. The host system can program any value greater than or equal to 0x08 into RxEarlyThresh. The ST201 needs a minimum of 8 frame bytes to perform destination address filtering before generating an RxEarly interrupt. The value in RxEarlyThresh may also determine how many bytes of a frame must be received before RxDMA transfers for the frame are allowed to begin. If RxEarlyEnable in DMACtrl is set, a frame becomes eligible to start RxDMA when RxEarlyThresh bytes have been received. Setting RxEarlyThresh too low will cause the host to respond to the interrupt before the entire receive frame header has been received. Setting RxEarlyThresh too high will introduce unnecessary delays in the system's receive response sequence. When RxEarlyThresh is set to a value that is greater than the length of the received frame, a RxComplete interrupt will occur at the completion of frame reception rather than a RxEarly interrupt. If the host system is particularly slow in responding to a RxEarly interrupt, then it is likely that the frame will have been completely received by the time the driver examines the ST201. In this case, RxEarly will be overridden by RxComplete. RxEarly is cleared when RxComplete becomes set, hence they are mutually exclusive. In order to prevent spurious interrupts, RxComplete should only be disabled if RxEarly is also disabled.

In some host systems, it may be desirable to copy received frame data out of the scatter buffer to the protocol buffer while the frame is still being transferred by RxDMA. The RxDMAStatus register is provided for this purpose. If the host system sets the RxDMAHalt bit in the DMACtrl register, reads the RxDMAListPtr register and the RxDMAStatus register, then sets the RxDMAResume bit in the DMACtrl register, the host system can determine how much of the frame has been transferred by RxDMA. The RxDMAStatus register indicates the number of bytes transferred by RxDMA for the current RFD pointed to by the RxDMAListPtr register. The host system can then perform memory copies out of the RFD buffer in parallel with the RxDMA operation.

INTERRUPTS

The term "interrupt" is used loosely to refer to interrupts and indications. An interrupt is the actual assertion of the hardware interrupt signal on the PCI bus. An indication, or a set bit in the IntStatus register, is the reporting of any event enabled by the host. The host system will configure the ST201 to generate an interrupt for any indication that is of interest to it. There are 10 different types of interrupt indications that can be generated by the ST201. The IntEnable register controls which of the 10 indication bits can assert a hardware interrupt. In order for an indication bit to be allowed to generate an interrupt, its corresponding bit-position in IntEnable must be set. When responding to an interrupt, the host reads the IntStatus register to determine the cause of the interrupt. The least significant bit of IntStatus, InterruptStatus, is always set whenever any of the interrupts are asserted. InterruptStatus must be explicitly acknowledged (cleared) by writing a 1 into the bit in order to prevent spurious interrupts on the host bus.

Interrupts are acknowledged by the host carrying out various actions specific to each interrupt. These actions are as follows:

- HostError, acknowledged by issuing the appropriate resets
- TxComplete, acknowledged by writing to TxStatus
- RxComplete, acknowledged automatically by the hardware
- UpdateStats, acknowledged by reading statistics registers
- InterruptStatus, acknowledged by writing a 1 into this bit
- RxEarly, acknowledged by writing a 1 into this bit
- IntRequested, acknowledged by writing a 1 into this bit
- LinkEvent, acknowledged by writing a 1 into this bit
- TxDMAComplete, acknowledged by writing a 1 into this bit
- RxDMAComplete, acknowledged by writing a 1 into this bit
- MACControlFrame, acknowledged by writing a 1 into this bit

STATISTICS

The ST201 implements 16 statistics counters of various widths. Each statistic implemented complies to the corresponding definition given in the IEEE 802.3 standard. Setting the StatisticsEnable bit in the MACCtrl register enables the gathering of statistics. Reading a statistics register will clear the read register. Statistic registers may be read without disabling statistics gathering. For diagnostics and testing purposes, the host system may write a value to a statistic register, in which case the value written is added to the current value of the register. Whenever one or more of the statistics registers reaches 75% of its maximum value, an UpdateStats interrupt is generated. Reading that statistics register will acknowledge the UpdateStats interrupt. A summary of the transmit and receive statistics follows. Detailed descriptions of the statistic registers related to data transmission and reception can be found in the Registers and Data Structures section.

TRANSMIT STATISTICS

- FramesTransmittedOk: The number frames of all types transmitted without errors. Loss of carrier is not considered to be an error by this statistic.
- BroadcastFramesTransmittedOk: The number of frames with broadcast destination address that are transmitted without errors.
- MulticastFramesTransmittedOk: The number of frames with multicast destination address that are transmitted without errors.
- OctetsTransmittedOk: The number of total octets for all frames transmitted without error.
- FramesWithDeferredXmission: A count of frames whose transmission was delayed on it's first attempt because network traffic.
- FramesWithExcessiveDeferral: If the transmission of a frame has been deferred for an excessive period of time due to network traffic, the event is recorded in this statistic.
- SingleCollisionFrames: Frames that are transmitted without errors after one and only one collision (including late collisions) are counted by this register.
- MultipleCollisionFrames: All frames transmitted without error after experiencing from 2 through 15 collisions (including late collisions) are counted here.
- LateCollisions: Every occurrence of a late collision (there could be more than one per frame transmitted) is counted by this statistic.
- FramesAbortedDueToXSColls: If the transmission of a frame had to be aborted due to excessive collisions, the event is recorded in this statistic.
- CarrierSenseErrors: Frames that were transmitted without error but experienced a loss of carrier are counted by this statistic.

RECEIVE STATISTICS

- FramesReceivedOk: Frames of all types that are received without error are counted here.
- BroadcastFramesReceivedOk: Frames of broadcast destination address that are received without error are counted here.
- MulticastFramesReceivedOk: Frames of multicast destination address that are received without error are counted here.
- OctetsReceivedOk: A total octet count for all frames received without error.
- FramesLostRxErrors: This is a count of frames that would otherwise be received by the ST201, but could not be accepted due to an overrun condition in the RxFifo.

PCI BUS MASTER OPERATION

The ST201 supports all of the PCI memory commands and decides on a burst-by-burst basis which command to use in order to maximize bus efficiency. The list of PCI memory commands is shown below. For all commands, "read" and "write" are with respect to the ST201 (i.e. read implies the ST201 obtains information from an off-chip location, write implies the ST201 sends information to an off-chip location).

- Memory Read (MR)
- Memory Read Line (MRL)
- Memory Read Multiple (MRM).
- Memory Write (MW)
- Memory Write Invalidate (MWI)

MR is used for all fetches of descriptor information. For reads of transmit frame data, MR, MRL, or MRM is used, depending upon the remaining number of bytes in the fragment, the amount of free space in the TxFIFO, and whether the RxDMA Logic is requesting a bus master operation.

MW is used for all descriptor writes. Writes of receive frame data use either MW or MWI, depending upon the remaining number of bytes in the fragment, the amount of frame data in the RxFIFO, and whether the TxDMA Logic is requesting a bus master operation.

The ST201 provides three configuration bits to control the use of advanced memory commands. The MWIEnable bit in the ConfigCommand configuration register allows the host to enable or disable the use of MWI. The MWIDisable and MRLDisable bits in DMACtrl allow the host system the ability to

disable the use of MWI and MRL. MWIDisable and MRLDisable are cleared by default, enabling MWI and MRL.

The ST201 provides a set of registers that control the PCI burst behavior. These registers allow a trade-off to be made between PCI bus efficiency and under run/overrun frequency. Arbitration logic within the PCI Bus Interface block accepts bus requests from the TxDMA Logic and RxDMA Logic. The TxDMA Logic uses the TxDMABurstThresh register, as described in the TxDMA Logic section, to delay the bus request until there is enough free space in the TxFIFO for a long, efficient burst. The TxDMA Logic can also make an urgent bus request as described in the TxDMA Logic section, where burst efficiency is sacrificed in favor of avoiding a TxFIFO under run condition.

The RxDMA process is described in the RxDMA Logic section. Typically, RxDMA requests will be forwarded to the Arbiter, however RxDMA Urgent Requests are also possible in order to prevent receive overruns. The Arbiter services the four requests in the fixed priority order as described in the PCI Bus Interface section.

POWER MANAGEMENT

The ST201 supports operating system directed power management according to the ACPI specification. Power management registers in the PCI configuration space, as defined by the PCI Bus Power Management Interface specification, Revision 1.0 are described in the Registers and Data Structures section.

The ST201 supports several power management states. The PowerState field in the PowerMgmtCtrl register determines ST201's current power state. The power states are defined as follows:

- D0 Uninitialized (power state 0) is entered as a result of hardware reset, or after a transition from D3 Hot to D0. This state is the same as D0 Active except that the PCI configuration registers are uninitialized. In this state, the ST201 responds to PCI configuration cycles only.
- D0 Active (power state 0) is the normal operational power state for the ST201. In this state, the PCI configuration registers have been initialized by the system, including the IoSpace, MemorySpace, and BusMaster bits in Config-Command, so the ST201 is able to respond to PCI I/O, memory and configuration cycles and can operate as a PCI master. The ST201 cannot signal wake (PMEN) from the D0 state.
- D1 (power state 1) is a "light-sleep" state. The ST201 optionally supports this state determined by the D1Support bit in the ConfigParm word in EEPROM. The D1 state allows transition back to D0 with no delay. In this state, the ST201 responds to PCI configuration accesses, to allow the system to change the power state. In D1 the ST201 does not respond to any PCI I/O or memory accesses. The ST201's function in the D1 state is to recognize wake events and link state events and pass them on to the system by asserting the PMEN signal on the PCI bus.
- D2 (power state 2) is a partial power-down state. The ST201 optionally supports this state determined by the D2Support bit in the ConfigParm word in EEPROM. D2 allows a faster transition back to D0 than is possible from the D3 state. In this state, the ST201 responds to PCI configuration accesses, to allow the system to change the power state. In D2 the ST201 does not respond to any PCI I/O or memory accesses. The ST201's function in the D2 state is to recognize wake events and link state events and pass them on to the system by asserting the PMEN signal on the PCI bus.
- D3 Hot (power state 3) is the full power-down state for the ST201. In D3 Hot, the ST201 loses all PCI configuration information except for the value in PowerState. In this state, the ST201 responds to PCI configuration accesses, to allow the system to change the power state back to D0 Uninitialized. In D3 hot, the ST201 does not respond to any PCI I/O or memory accesses. The ST201's main responsibility in the D3 Hot state is to recognize wake events and link state events and signal those to the system by asserting the PMEN signal on the PCI bus.
- D3 Cold (power state undefined) is the power-off state for the ST201. The ST201 does not function in this state. When power is restored, the system guarantees the assertion of hardware reset, which puts the ST201 into the D0 Uninitialized state.

The ST201 can generate wake events to the system as a result of Wake Packet reception, Magic Packet reception, or due to a change in the link status. The WakeEvent register gives the host system control over which of these events are passed to the system. Wake events are signaled over the PCI bus using the PMEN pin.

A Wake Packet event is controlled by the WakePktEnable bit in WakeEvent register. WakePktEnable has no effect when ST201 is in the D0 power state, as the wake process can only take place in states

D1, D2, or D3. When the ST201 detects a Wake Packet, it signals a wake event on PMEN (if PMEN assertion is enabled), and sets the WakePktEvent bit in the WakeEvent register. The ST201 can signal that a wake event has occurred when it receives a pre-defined frame from another station. The host system transfers a set of frame data patterns into the TxFIFO using the TxDMA function before placing the ST201 in a power-down state. Once powered down, the ST201 compares receive frames with the frame patterns in the TxFIFO. When a matching frame is received (and also passes the filtering mode set in the ReceiveMode register), a wake event is signaled.

Frame patterns are written to the TxFIFO in a single "pseudo-packet". Prior to transferring this pseudo-packet, the host system should first issue TxReset (to reset the TxFIFO pointers and prevent transmission) then prepare a TFD that points to a single data buffer. The buffer should contain one or more frame patterns placed contiguously. The number of frame patterns is limited by the TxFIFO size. The TxDMAFragLen field in the TFD must exactly equal the sum of the frame pattern bytes. Also, the host system must set WordAlign to 'x1' in the TransmitFrameControl field of the TFD to prevent frame word-alignment. Finally, the host system must write the TFD's address to the TxDMAListPtr register to transfer the frame into the TxFIFO.

The frame patterns in the TxFIFO specify which bytes in the incoming frames are to be examined. A CRC is calculated over these bytes and compared with a CRC value supplied in the frame pattern. This matching technique may result in false wake events being reported to the host system. Each wake packet pattern contains one or more byte-offset/byte-count pairs, an end-of-pattern symbol, and a 4-byte CRC value. The byte-offset indicates the number of frame bytes to be skipped in order to reach the next group of bytes to be included in the CRC calculation. The byte-count indicates the number of bytes in the next group to be included in the CRC calculation. End-of pattern, which is a byte value of 00, indicates the end of the pattern for that wake frame. Immediately following the end-of-pattern is a 4-byte CRC. The CRC calculation uses the same polynomial as the Ethernet MAC FCS. The pseudo packet frame patterns are described in the Registers and Data Structures section.

An example pseudo-packet (based on the ARP packet example from Appendix A of the "OnNow Network Device Class Power Management Specification") loaded into the TxFIFO of the ST201 is shown in Figure 6.

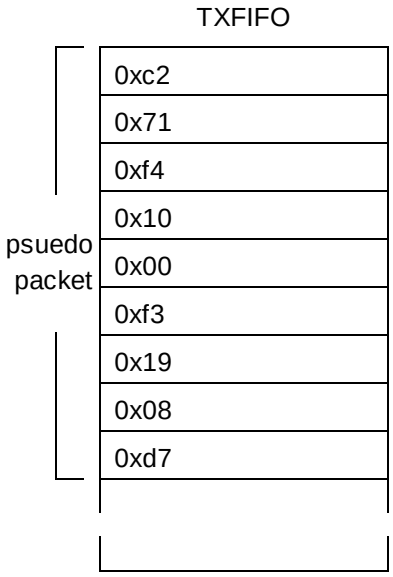


FIGURE 6: Example Psuedo Packet

Using the pseudo packet in Figure 6, the ST201 will assert a wake event if a packet of the form shown in Figure 7 is received whereby a 32-bit CRC over the indicated bytes of the received packet yields the value 0xf31908d7.

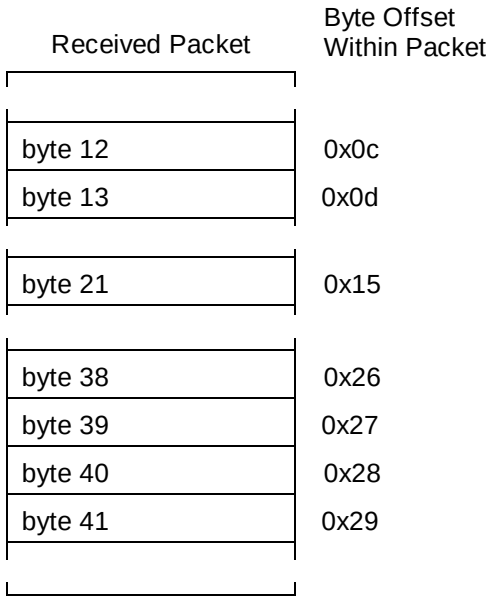


FIGURE 7: Example Wake Packet

The ST201 also supports Magic Packet™ technology developed by Advanced Micro Devices to allow remote wake-up of a sleeping station on a

network via transmission of a special frame. Once the ST201 has been placed in Magic Packet mode and put to sleep, it scans all incoming frames addressed to it for a data sequence consisting of 16 consecutive repetitions of its own 48-bit Ethernet MAC StationAddress. This sequence can be located anywhere within the frame, but must be preceded by a synchronization stream. The synchronization stream is defined as 6 bytes of 0xFF. For example, if the MAC address programmed into the StationAddress register is 0x11:22:33:44:55:66, then the ST201 would be scanning for the frame data shown in Figure 8.

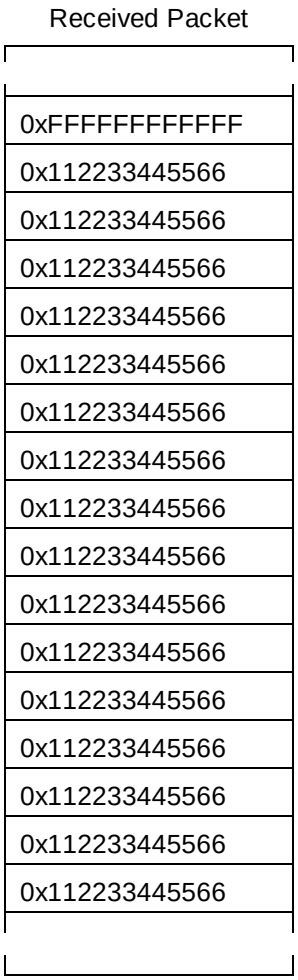


FIGURE 8: Example Magic Packet

Magic Packet wake up is controlled by the MagicPktEnable bit in the WakeEvent register. A wake event can only take place in the D1, D2, or D3 states, and MagicPktEnable has no effect when the ST201 is in the D0 power state. The Magic Packet must also pass the address matching criteria set in ReceiveMode. A Magic Packet may also be a broadcast frame. When the ST201 detects a Magic Packet, it signals a wake event on PMEN (if PMEN assertion is enabled), and sets the MagicPktEvent bit in WakeEvent.

The ST201 can also signal a wake event when it senses a change in the network link state, either from LINK_OK to LINK_FAIL, or vice versa. Link state wake is controlled by the LinkEventEnable bit in the WakeEvent register. At the time LinkEventEnable is set by the host system, the ST201 samples the current link state. It then waits for the link state to change. If the link state changes before the ST201 returns to state D0 or LinkEventEnable is cleared, LinkEvent is set in WakeEvent, and (if it is enabled) the PMEN signal is asserted.

HOST SYSTEM RELATED INFORMATION

PROGRAMMING THE MII MANAGEMENT INTERFACE

Register accesses across the MII Management Interface occur serially, and are controlled by manipulating the MgmtClk, MgmtData, and MgmtDir bits in the PhyCtrl register. The direction of the serial transmission is controlled by MgmtDir. MgmtDir is set when writing bits to the PHY device, and cleared when reading bits from the PHY. Data bits are read from and written to MgmtData. MgmtClk supplies the synchronization clock for the interface.

Management frames are transmitted from the ST201 to the PHY device using one or more Read, Write, or Z cycles. The actions which must be taken by the host system to perform a Read, Write, or Z cycle are described below. Note, in the cycle descriptions below, time delays of 200ns can generally be assumed from back-to back I/O cycles on the PCI bus. The host system may also use an arbitrarily long timer to guarantee meeting minimum transition delays.

To perform a Read cycle, consisting of a single data bit read from the PHY device, the host system must follow the procedure below.

1. Clear MgmtClk
2. Wait a minimum of 200 ns
3. Set MgmtClk
4. Wait a minimum of 200 ns
5. Read the next data bit from MgmtData
6. Wait a minimum of 200 ns

To perform a Write cycle, consisting of a single data bit written to the PHY device, the host system should follow the procedure below.

1. Clear MgmtClk
2. Wait a minimum of 200 ns

3. Set MgmtClk
4. Write the desired data bit to MgmtData
5. Wait a minimum of 200 ns

To perform a Z cycle used during the Turnaround portion of a register read frame, the host system should follow the procedure below.

1. Clear MgmtClk
2. Wait a minimum of 200 ns
3. Set MgmtClk
4. Clear MgmtDir
5. Wait a minimum of 200 ns

Using the read, write, and Z cycle procedures, a Read management frame is executed as follows.

1. Set MgmtDir.
2. Execute 32 write cycles to transmit the 32 bit Preamble of 0xffffffff.
3. Execute 2 write cycles to transmit the 2 bit Start of Frame of 0x01.
4. Execute 2 write cycles to transmit the 2 bit Operation Code of 0x02.
5. Execute 5 write cycles to transmit the 5 bit PHY Address to identify the target PHY device for the Read frame.
6. Execute 5 write cycles to transmit the 5 bit Register Address to identify the register to be read within the PHY device.
7. Execute a Z cycle to prepare the interface to receive read data bits (first half of Read Turnaround).
8. Execute a single read cycle (second half of Read Turnaround). The bit read from the PHY will be a zero if the PHY intends to respond to the Read frame. A value of one indicates that no PHY device is responding, and the data to follow is invalid.
9. Execute 16 read cycles to read the data from the PHY register. Data bits are read starting with register bit 15 and ending with register bit 0.
10. Execute a Z cycle to terminate the frame.

Using the read, write, and Z cycle procedures, a Write management frame is executed as follows.

1. Set MgmtDir.
2. Execute 32 write cycles to transmit the 32 bit Preamble of 0xffffffff.
3. Execute 2 write cycles to transmit the 2 bit Start of Frame of 0x01.
4. Execute 2 write cycles to transmit the 2 bit Operation Code of 0x01.
5. Execute 5 write cycles to transmit the 5 bit PHY Address to identify the target PHY device for the Write frame.
6. Execute 5 write cycles to transmit the 5 bit Register Address to identify the register to be written within the PHY device.
7. Execute 2 write cycles to transmit the 2 bit Write Turnaround value of 0x02.
8. Execute 16 write cycles to write the data to the PHY register. Data bits are written starting with register bit 15 and ending with register bit 0.
9. Execute a Z cycle to terminate the frame.

EEPROM COMMANDS

The EepromCtrl register provides the host with a method for issuing commands to the ST201's serial EEPROM controller. Individual 16-bit word locations within the EEPROM may be written, read or erased. Also, the EEPROM's WriteEnable, WriteDisable, EraseAll and WriteAll commands can be issued.

Two-bit opcodes and 8-bit addresses are written to the EepromCtrl register to cause the ST201 to carry out the desired EEPROM command. If data is to be written to the EEPROM, the 16-bit data word must be written to EepromData by the host system prior to issuing the associated write command. Similarly, if data is to be read from the EEPROM, the read data will be available via EepromData register after issuing the associated read command.

A mechanism within the EEPROM interface automatically disables writes and erasures to prevent accidental data changes should power be interrupted. The ST201 disables writes and erasures after every write or erase type command has been executed. To write or erase a series of locations, the host must issue the WriteEnable command prior to every write or erase type command.

The serial EEPROM can only clear bits to zero during a write command and cannot set individual bits to ones. Therefore, an Erase or EraseAll command must be issued prior to attempting to write data to the EEPROM.

The EEPROM is a particularly slow device. It is important that the host wait until the EepromBusy bit is false before issuing a command to EepromCtrl.

The procedure for a typical write operation to the EEPROM is as follows:

1. Verify EepromBusy is false.
2. Issue the WriteEnable command (opcode = 00 11xx xxxx)
3. Verify EepromBusy is false.
4. Issue EraseRegister command (opcode = 11 aaaa aaaa)
5. Verify EepromBusy is false.
6. Write data pattern to EepromData.
7. Issue WriteEnable command (opcode = 00 11xx xxxx)

8. Verify EepromBusy is false.
9. Issue WriteRegister command (opcode = 01 aaaa aaaa)

Step 4 through 8 may be skipped for certain types of EEPROM devices.

ADAPTER TXDMA SEQUENCE

Beginning with the host system writing to the TxDMAListPtr register (when starting from an empty TxDMAList, for instance), the ST201 performs the following procedure during transfers of TxDMA frames.

1. Verifies the TxDMAListPtr is non-zero.
2. Verifies not in the TxDMAHalt state.
3. Fetches the second dword from the TFD pointed to by TxDMAListPtr and writes this value into the TxFIFO.
4. One by one, fetches the TxDMAFragAddr/TxDMAFragLen entries from the TFD, and moves the associated data fragments to the TxFIFO.
5. Sets the TxDMAComplete bit in the TFD.
6. If TxDMAHalt is in effect, waits until a TxDMAResume is issued.
7. If a transmit under run has occurred, waits until the host system sets the TxReset bit.
8. Re-fetches the TFC and, if the TxDMAIndicate bit is set, sets TxDMAComplete (which may in turn cause an interrupt if the IntEnable masks are set appropriately)
9. Fetches the TxDMANextPtr from the current TFD. If TxDMANextPtr is zero, the TxDMA Logic becomes idle. If polling is disabled (TxDMAPollPeriod is zero), the TxDMA Logic waits for a non-zero value to be written to TxDMAListPtr. If polling is enabled (TxDMAPollPeriod is non-zero), the old value in TxDMAListPtr is preserved, and the ST201 polls on TxDMANextPtr in the TFD until it fetches a non-zero value from it. If the value fetched from TxDMANextPtr is non-zero, then the value is loaded into TxDMAListPtr, advancing the ST201 to the new TFD.
10. With a new TFD to process, the ST201 returns to step 2.

ADDING TFD'S TO THE END OF THE TXDMALIST

The following sequence describes the process for adding TFD's to the end of the TxDMAList when TxDMA polling is disabled (TxDMAPollPeriod is zero).

1. Set the TxDMAHalt bit.
2. Wait for the TxDMAHalt to complete by polling on TxDMAHalted bit.
3. Update TxDMANextPtr in the last TFD in the TxDMAList with the address of the TFD being added.
4. Read TxDMAListPtr.
5. If TxDMAListPtr is zero, write the address of the new TFD to TxDMAListPtr.
6. Resume the TxDMA Logic by setting TxDMAResume bit. The TxDMA Logic will become idle when it fetches a zero TxDMANextPtr value from a TFD. One way to restart the TxDMA process is by writing a non-zero value to TxDMAListPtr.

When polling is enabled (TxDMAPollPeriod is non-zero), TFD's can be added to the end of the TxDMAList with no register accesses by writing the address of the new TFD in the last TFD's TxDMANextPtr field.

INSERTING A TFD AT THE HEAD OF THE TXDMALIST

TFD's cannot be added before the active TFD in the TxDMAList, they can only be added after the active (unfinished) TFD. The following sequence describes the process for adding TFD's to the head of the TxDMAList when TxDMA polling is disabled (TxDMAPollPeriod is zero).

1. Set TxDMAHalt bit.
2. Wait for the TxDMAHalt to complete by polling the TxDMAHalted bit. When halted, the TxDMAListPtr register will hold the address of the TFD which just completed transfer into the ST201 (the "last TFD in the TxDMAList").
3. Find the last TFD in the TxDMAList, corresponding to the TFD at the address held in the TxDMAListPtr register. This will also be the last TFD in the TxDMAList with the TxDMAComplete bit set.
4. Copy the value in the "last TFD's" TxDMANextPtr into the TxDMANextPtr field of the TFD to be inserted.
5. Update TxDMANextPtr field in the "last TFD" with the address of the inserted TFD.
6. Read TxDMAListPtr.
7. If TxDMAListPtr is zero, write the address of the inserted TFD to TxDMAListPtr.
8. Resume the TxDMA Logic by setting TxDMAResume.

When polling is enabled (TxDMAPollPeriod is non-zero), TFD's can be inserted to the head of the TxDMAList with no register accesses as follows.

1. Find the first TFD in the list with TxDMAComplete bit cleared (the "first TFD").
2. Set the "first TFD's" TxDMANextPtr to zero.
3. Check the TxDMAComplete bit of "the first TFD". If clear, proceed to the next step. If set, it's too late to insert the new TFD at the loca-

tion of the “first TFD” in the TxDMAList. Restore the TxDMANextPtr of the “first TFD”, and restart this process.
4. Copy the value of the “first TFD’s” TxDMANextPtr into the TxDMANextPtr field of the inserted TFD.
5. Update the TxDMANextPtr field of the “first TFD” with the address of the inserted TFD.

TRANSMIT INTERRUPT OPTIMIZATIONS

The transmit mechanism can be optimized by the host system, allowing a reduction in the number of interrupts generated. The host system can limit the number of frames in the TxDMAList for which a TxDMAComplete interrupt is generated. For example, the host system could only set TxDMAIndicate for the frame on the tail of the list (clearing TxDMAIndicate for the current tail before adding a new frame to the list). Or it might require an interrupt every N frames. In any case, on each interrupt it would then dequeue all of the frames which were transferred via TxDMA before that interrupt occurred (TFDs in which TxDMAComplete is set). Obviously, the host system is responsible for the trade-off between latency and the number of interrupts generated by the ST201.

RXDMA SEQUENCE

The ST201 performs the following procedure during transfers of RxDMA frames.
1. Verifies the RxDMAListPtr register is non-zero.
2. Verifies not in the RxDMAHalt state.
3. Resets the RxDMAStatus register.
4. Fetches ReceiveFrameStatus from the current RFD. If the current RFD has the RxDMAComplete bit set, the ST201 performs an implicit RxDMAHalt, halting the RxDMA process. (If RxDMAPollPeriod contains a nonzero value, the ST201 will then poll on the RxDMAComplete bit, waiting for it to be cleared before continuing.) Otherwise, the RxDMA process continues.
5. Waits for the top receive frame to become eligible for RxDMA.
6. Transfers the frame via RxDMA into the fragments specified in the RFD, or into the implied buffer if ImpliedBufferEnable is set. If there is more data in the frame than space in the fragment buffers, the ST201 generates a RxDMAOverflow error.
7. As the frame is being transferred by the RxDMA process, the ST201 maintains the RxDMAStatus register, specifically the RxDMAFrameLen field.
8. At the end of the frame RxDMA process, the ST201 updates the RxDMAStatus register with any error codes from the frame transfer and sets the RxDMAComplete bit.
9. Issues an internal RxDiscard and waits for completion.
10. If a RxDMAHalt has been asserted, waits until a RxDMAResume has been issued.
11. Fetches RxDMANextPtr from the RFD. If RxDMANextPtr is zero and polling is enabled, the ST201 begins a polling loop. If polling is disabled, the ST201 loads the fetched value into RxDMAListPtr.
12. Writes ReceiveFrameStatus to the RFD in host memory.
13. If a polling loop has begun, polls on RxDMANextPtr until a non-zero value is fetched. Loads the value into RxDMAListPtr.
14. If the RxDMAListPtr value is zero (polling is disabled), then the RxDMA Logic becomes idle, waiting for a non-zero value to be written into the RxDMAListPtr register.
15. Repeat process at step 1.

WAKE EVENT PROGRAMMING

This section describes the sequences involved in programming ST201 for wake events.

Powering down the ST201 will typically occur while in the operating state (D0). The host system is notified by the operating system that a power state change is imminent. The host system prepares for power down with these steps.
1. Halt the ST201 TxDMA process, halt the TxDMA Logic, wait for any TxDMA in progress to complete, wait for any transmissions to complete, and issue TxReset to reset the TxFIFO pointers.
2. Perform the RxDMA process for any receive frames remaining in RxFIFO, halt RxDMA Logic (frames will potentially keep filling the RxFIFO between now and when operating system powers down the system, and once the power down state is entered and Wakeup Packet scanning enabled, these frames in the RxFIFO will be scanned and could potentially wake the system immediately).
3. Clear IntEnable register so no interrupts occur before PowerState is changed.
4. Save any volatile state, such as a pending power state and the HashTable settings, to system memory (system memory is restored after a power down).
5. The host system transfers Wakeup Packet patterns to the TxFIFO (if Wakeup Packets are to be enabled) and programs the WakeEvent register to enable the desired wake events. The

host system then returns to the operating system an indication of readiness to be powered down (making sure to leave the ReceiveMode register set to receive the appropriate Wake/Magic packets). The operating system eventually writes to the PowerMgmtCtrl register, placing the ST201 in one of the power down states, and enabling PMEN assertion, while the ST201 monitors for the occurrence of enabled wake events.

WAKE EVENT

When a desired wake event occurs, the ST201 sets the appropriate event bit in the WakeEvent register, sets the PmeStatus bit in the PowerMgmtCtrl register, and asserts the PMEN pin.

The host system responds to PMEN by scanning the power management configuration registers of all devices, looking for the device which asserted PMEN. If the device with the ST201 signaled wake, the system will find PmeStatus set in ST201's PowerMgmtCtrl register. The operating system then clears the PmeEn bit in the PowerMgmtCtrl register causing PMEN to be de-asserted.

The operating system raises the power state (probably to D0) by writing to the PowerState bits in the PowerMgmtCtrl register. If the ST201 was previously in the D3 state, PCI configuration is lost and must be restored by the operating system.

The host system must set TxReset to clear any wake patterns out of the TxFIFO (if this is not done, the patterns will be treated as frames and transmitted once the transmitter is enabled).

The host system reads the WakeEvent register to determine the wake event, and if requested, passes it back to the operating system. The host system restores any volatile state that was saved in the power down sequence. The host system reenables interrupts by programming IntEnable. The host system restores the RxDMAList (and any other data structures required for operation). Any wake packets in the RxFIFO are transferred by RxDMA and passed to the operating system.

REGISTERS AND DATA STRUCTURES

DMA DATA STRUCTURES

A TFD is used to move data, which is to be transmitted onto a LAN, from host system memory to the TxFIFO within the ST201. A TFD is 16 to 512 bytes in length, and it's location in host system memory is indicated by the value in the TxDMAListPtr register.

A RFD is used to move data, which has been received from a LAN, from the RxFIFO within the ST201 to host system memory. A RFD is 16 to 512 bytes in length, and it's location in host system memory is indicated by the value in the RxDMAListPtr register.

Figure 9 shows the two DMA data structures.

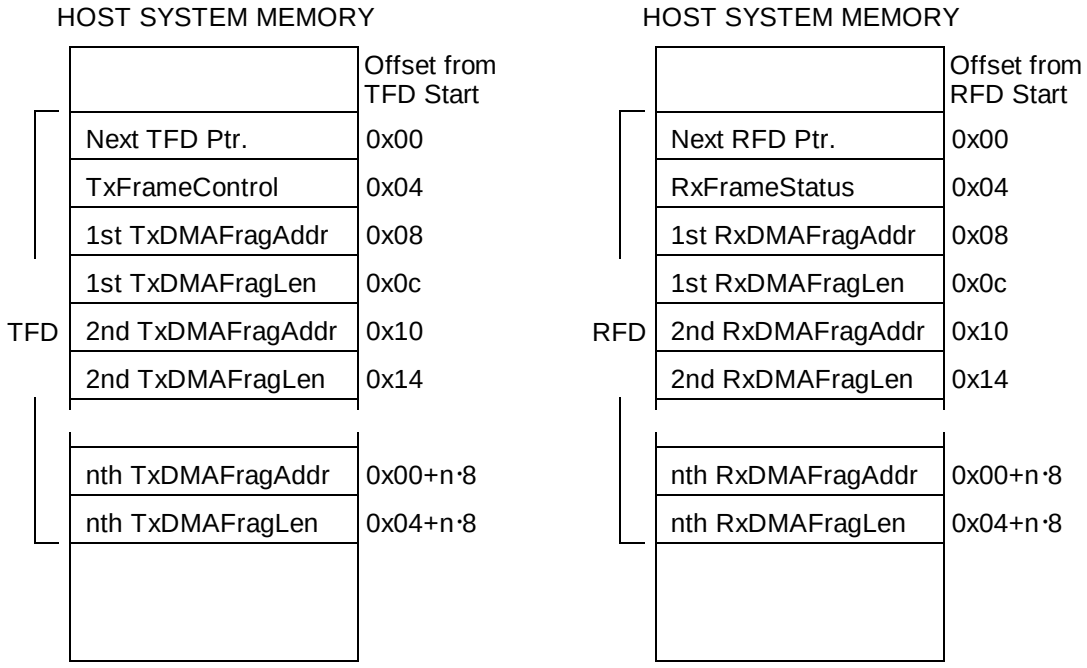


FIGURE 9: TFD and RFD DMA Data Structures

TXDMAFRAGADDR

Class.....DMA Data Structures, TFD

Base Address .....Start of TFD

Address Offset.....0x00+n·8 for nth fragment

Access Mode.....Read/Write

Width .....32 bits

BIT	BIT NAME	BIT DESCRIPTION
31..0	TxDMAFragAddr	Transmit Fragment Address contains the physical address of a contiguous block of data to be transferred by TxDMA into the ST201 and transmitted. A fragment can start on any byte boundary.

TXDMAFRAGLEN

Class.....DMA Data Structures, TFD

Base Address .....Start of TFD

Address Offset.....0x04+n·8 for nth fragment

Access Mode.....Read/Write

Width .....32 bits

Transmit Fragment Length (TxDMAFragLen) contains fragment length and control information for the block of data pointed to by the corresponding TxDMAFragAddr.

BIT	BIT NAME	BIT DESCRIPTION
12..0	FragLen	The length of the contiguous block of data pointed to by the TxDMA-FragAddr. The maximum fragment length is 8192 bytes.
30..13	Reserved	Reserved for future use. Should be set to 0.
31	TxDMAFragLast	Set by the host system to indicate the last fragment of the transmit frame and that the ST201 should proceed to the next TFD.

TXDMANEXTPTR

Class.....DMA Data Structures, TFD

Base Address .....Start of TFD

Address Offset.....0x00

Access Mode.....Read/Write

Width .....32 bits

BIT	BIT NAME	BIT DESCRIPTION
31..0	TxDMANextPtr	Transmit Next Pointer, the first double word in the TFD contains the physical address of the next TFD in the TxDMAList. The value of zero accompanies the last frame of the list and it indicates there are no more TFD's in the TxDMAList. All TFD's must be aligned on 8-byte physical address boundaries, requiring Bits [2..0] of TxDMANextPtr to be zero.

TXFRAMECONTROL

Class.....DMA Data Structures, TFD

Base Address .....Start of TFD

Address Offset.....0x04

Access Mode.....Read/Write

Width .....32 bits

TxFrameControl contains frame control information for the TxDMA function and the transmit function.

BIT	BIT NAME	BIT DESCRIPTION
1..0	WordAlign	These bits determine the boundary to which transmit frame lengths are rounded up in the TxFIFO, and transmitted onto the network medium. 00: Align to dword 10: Align to word X1: Align disabled
9..2	FrameId	This field can be used as a frame ID or sequence number. This value is saved with the frame in the TxFIFO, and made visible in the TxFrameId register while the frame is being transmitted. When a transmit error occurs, the driver checks TxFrameId to determine which frame experienced the error.
12..10	Reserved	Reserved for future use. Should be set to 0.
13	FcsAppendDisable	The host system sets this bit to prevent the ST201 from appending the 4-byte FCS to the end of the current frame. In this case, the host system must supply the frame's FCS as part of the data transferred by TxDMA to the TxFIFO. An exception exists when a transmit under run occurs; in this case a guaranteed-bad FCS will be appended to the frame by the ST201. When FcsAppendDisable is cleared, the ST201 will compute and append FCS to this transmit frame.
14	Reserved	Reserved for future use. Should be set to 0.
15	TxIndicate	The host system sets this bit to request a TxComplete interrupt upon completion of MAC transmission of this frame. If TxComplete is cleared, no interrupt of transmit completion will be given by the ST201, unless a transmit error occurs.
16	TxDMAComplete	Indicates that the frame transfer by TxDMA is complete. The ST201 sets this bit after it has finished transferring via the TxDMA process all of the fragments specified in the TFD.
30..17	Reserved	Reserved for future use. Should be set to 0.
31	TxDMAIndicate	Set if the host system desires a TxDMAComplete interrupt upon completion of TxDMA of this frame. The TFC is read twice by the ST201; the first time to write the TFC to the TxFIFO before frame data transfer, and again after the TxDMA operation is complete to test TxDMAIndicate in order to determine whether to generate an interrupt. This allows the host system time to change TxDMAIndicate while the transfer by TxDMA is in progress.

RXDMANEXTPTR

Class.....DMA Data Structures, RFD

Base Address .....Start of RFD

Address Offset.....0x00

Access Mode.....Read/Write

Width .....32 bits

BIT	BIT NAME	BIT DESCRIPTION
31..0	RxDMANextPtr	The first dword in the RFD contains the physical address of the next RFD in the RxDMAList. If this is the last RFD in the RxDMAList, then this value must be zero. RFDs must be aligned on 8-byte physical address boundaries.

RXFRAMESTATUS

Class.....DMA Data Structures, RFD

Base Address .....Start of RFD

Address Offset.....0x04

Access Mode.....Read/Write

Width .....32 bits

The second dword in the RFD is ReceiveFrameStatus. At the end of a RxDMA frame transfer, the ST201 writes the value of the RxDMAStatus register into this location in the RFD. The bit definitions for TxFrameStatus for bits[31..29] and [27..0] are identical to the corresponding bits of the I/O Register RxDMAStatus. Only bit[28] differs between the RFD field RxFrameStatus and the register RxDMAStatus.

BIT	BIT NAME	BIT DESCRIPTION
12..0	RxDMAFrameLen	During frame RxDMA, RxDMAFrameLen gives a real-time indication of the number of bytes transferred by RxDMA for the frame. RxDMAFrameLen is cleared when the ST201 fetches a new RxDMAListPtr, and counts up in steps no larger than a bus master burst. When the frame has been completely transferred by RxDMA, RxDMAFrameLen indicates the true frame length, except in the case where the frame is larger than the number of bytes specified in the RxDMA fragments. In this case, the RxDMAOverflow bit will be set.
13	Reserved	Reserved for future use. Should be set to 0.
14	RxFrameError	Indicates that an error occurred in the receipt of the frame. The driver should examine bits 16 through 20 to determine the type of error(s). This bit is undefined until RxDMAComplete bit is set.
15	RxDMAComplete	Indicates that the frame transfer by RxDMA is complete. Unless a RxDMA halt is in effect this bit would normally only remain set momentarily (too short for the software to read it) since the hardware will then fetch the next RFD.
16	RxFIFOOverrun	Indicates that the hardware was unable to remove data from the RxFIFO quickly enough (most likely because the software failed to free a RFD quickly enough, or kept the ST201 in the RxDMAHalt state for too long). Bytes will be missing from the frame at one or more locations in the frame (unpredictable). This bit is undefined until RxDMAComplete bit is set.
17	RxRuntFrame	Indicates that the frame was a runt (less than 60 bytes). Normally such frames are not transferred by RxDMA unless RxEarlyThresh is set to a value less than the actual size of the runt frame, and the RxEarlyEnable of MacCtrl register must be set. This bit is undefined until RxDMAComplete bit is set.
18	RxAlignmentError	Indicates that the frame had an alignment error (bad FCS and dribble bits). This bit is undefined until RxDMAComplete bit is set.
19	RxFCSError	Indicates a FCS checksum error on the frame data. This bit is undefined until RxDMAComplete bit is set.
20	RxOversizedFrame	Indicates the frame size was equal to or greater than the value set in the MaxFrameSize register. This bit is undefined until RxDMAComplete bit is set.

BIT	BIT NAME	BIT DESCRIPTION
22..21	Reserved	Reserved for future use. Should be set to 0.
23	DribbleBits	Indicates that the frame had accompanying dribble bits. This bit is informational only, and does not indicate a frame error.
24	RxDMAOverflow	Indicates that the RFD had insufficient buffer space for the frame data and there were still data left to be transferred by RxDMA when the ST201 ran out of fragment space. The ST201 will transfer what it can into the buffers provided, discard the remainder of the frame and set this bit.
27..25	Reserved	Reserved for future use. Should be set to 0.
28	ImpliedBufferEnable	This bit enables a special RxDMA mode. Setting this bit instructs the ST201 not to fetch any RxDMAFragAddr/RxDMAFragLen entries from this RFD. Instead, the ST201 assumes there is one receive buffer of length 1528 bytes, starting immediately after ReceiveFrameStatus at (RFD address + 8). The host system sets this bit when it prepares the RFD. The ST201 tests this bit before RxDMA a frame, at the same time it tests the RxDMAComplete bit. When the ST201 updates ReceiveFrameStatus at the end of the RxDMA operation (in order to set RxDMAComplete), the value written to ImpliedBufferEnable is undefined. A driver cannot assume a certain value is left in this bit after the RFD is used. Therefore, the driver must write the desired value to this bit every time it releases a RFD to the ST201. This mode of operation reduces the number of information fetches by the ST201, and is intended for server applications in which frames are received into a ring of maximum frame sized buffers.
31..29	Reserved	Reserved for future use. Should be set to 0.

RXDMAFRAGADDR

Class.....DMA Data Structures, RFD

Base Address .....Start of RFD

Address Offset.....0x00+n·8 for nth fragment

Access Mode.....Read/Write

Width .....32 bits

BIT	BIT NAME	BIT DESCRIPTION
31..0	RxDMAFragAddr	The third and all subsequent odd dwords in the RFD contains the physical address of a contiguous block of system memory to which receive data is to be transferred by RxDMA. A fragment can start on any byte boundary.

RXDMAFRAGLEN

Class.....DMA Data Structures, RFD

Base Address .....Start of RFD

Address Offset.....0x04+n·8 for nth fragment

Access Mode.....Read/Write

Width .....32 bits

The fourth and all subsequent even dwords in the RFD contains fragment length and control information for the block of data pointed to by the previous RxDMAFragAddr.

BIT	BIT NAME	BIT DESCRIPTION
12..0	FragLen	The length of the contiguous block of data pointed to by the previous RxDMAFragAddr.
30..13	Reserved	Reserved for future use. Should be set to 0.
31	RxDMALastFrag	Set by the host system to indicate the last fragment of the receive frame.

WAKE EVENT DATA STRUCTURES

The first Wake Event Data Structure is the Pseudo Packet. A Pseudo Packet is a set of patterns loaded into the ST201 TxFIFO which specify bytes to be examined within received frames. A CRC is calculated over these bytes and compared with a CRC value supplied in the Pseudo Packet. If a match is found, the ST201 issues a Wake Event. The matching technique may result in false wake events being reported to the host system. Each Pseudo Packet consists of one or more byte-offset/byte-count pairs (or Pseudo Patterns), a terminator symbol, and a 4-byte CRC value. The byte offsets within the Pseudo Patterns indicate the number of received frame bytes to be skipped in order to reach the next group of bytes to be included in the CRC calculation. The byte-counts within the Pseudo Patterns indicate the number of bytes in the next group to be included in the CRC calculation. The terminator indicates the end of the Pseudo Patterns for the Pseudo Packet. Immediately following the terminator is a 4-byte CRC. If there is another Pseudo Packet, it will immediately follow the CRC value.

The second Wake Event Data Structure is the Magic Packet. Magic Packets are uniquely formatted frames, which upon reception invoke a Wake Event by the ST201. Once the ST201 has been placed in Magic Packet mode and put to sleep, it scans all incoming frames addressed to it for a data sequence consisting of a synchronization stream followed immediately by 16 consecutive repetitions of the station's own 48-bit Ethernet MAC station address. The sequence can be located anywhere within the received frame.

The pseudo packet and Magic Packet data structures are shown in Figure 10.

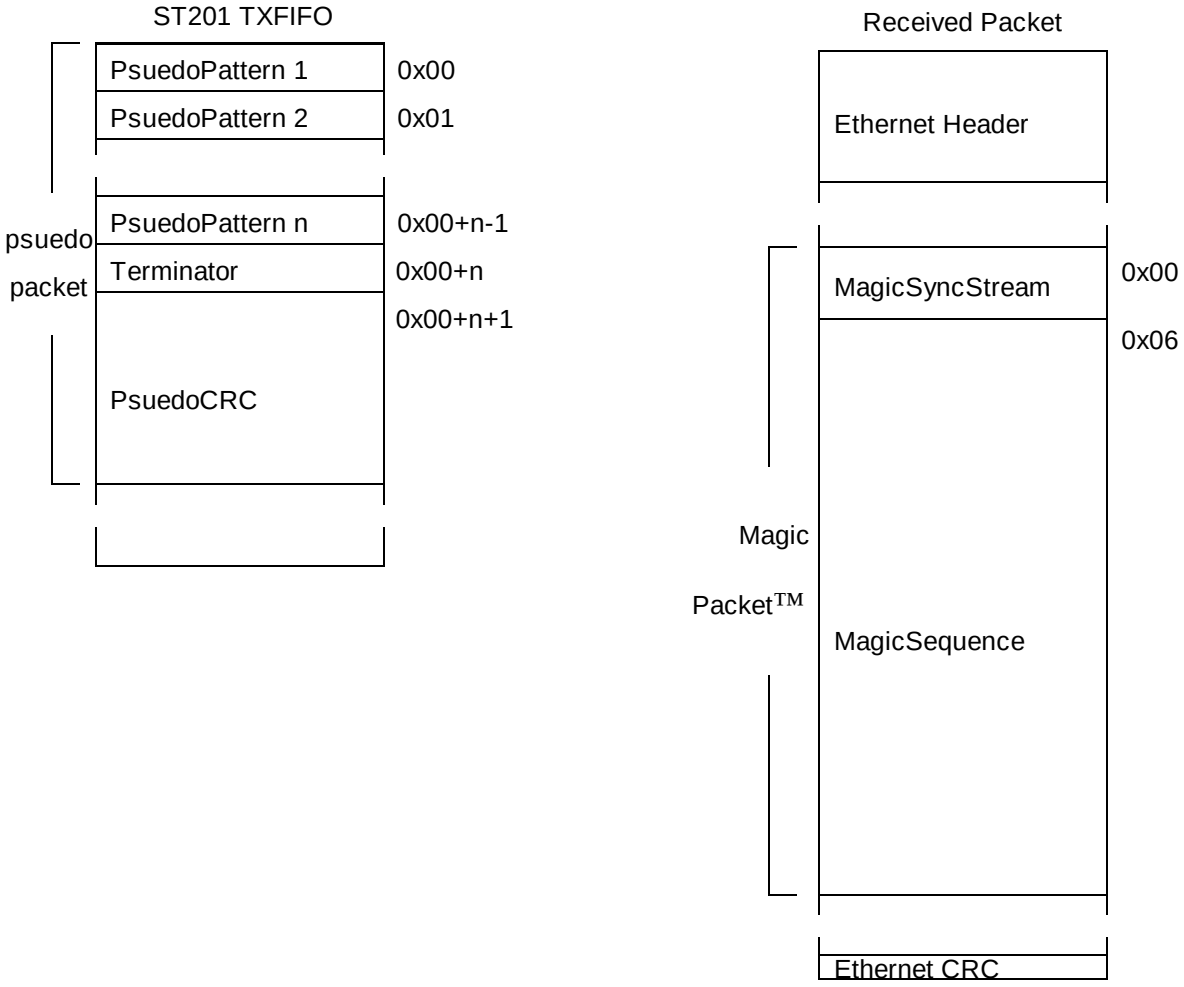


FIGURE 10: Wake Event Data Structures, Pseudo Packet and Magic Packet

PSEUDOPATTERN

Class.....Wake Event Data Structures, Pseudo Packet

Base Address .....Start of Pseudo Packet

Address Offset.....0x00 thru 0x00+n-1 for nth PseudoPattern

Access Mode.....Write only

Width .....8 bits

BIT	BIT NAME	BIT DESCRIPTION
3..0	ByteCount	ByteCount can take on a value of 0x0 to 0xe. A value of 0xf indicates an extended value. The extended value will occupy 8 bits and is contained in the next PseudoPattern. If both the ByteOffset and the ByteCount values are 0xf, the next PseudoPattern will be the extended ByteOffset, and the PseudoPattern after that will be the extended ByteCount.
7..4	ByteOffset	ByteOffset can take on a value of 0x0 to 0xe. A value of 0xf indicates an extended value. The extended value will occupy 8 bits and is contained in the next PseudoPattern. If both the ByteOffset and the ByteCount values are 0xf, the next PseudoPattern will be the extended ByteOffset, and the PseudoPattern after that will be the extended ByteCount.

TERMINATOR

Class.....Wake Event Data Structures, Pseudo Packet
Base Address .....Start of Pseudo Packet
Address Offset.....0x00+n for n PseudoPattern
Access Mode.....Write only
Width .....8 bits

BIT	BIT NAME	BIT DESCRIPTION
7..0	Terminator	A value of 0x00 indicates the end of the PseudoPattern.

PSEUDOCRC

Class.....Wake Event Data Structures, Pseudo Packet

Base Address .....Start of Pseudo Packet

Address Offset.....0x00+n+1 for n PseudoPatterns

Access Mode.....Write only

Width .....32 bits

The 32-bit CRC as defined in the IEEE 802.3 Ethernet standard for the FCS, taken over the bytes (indicated by the PseudoPattern values) of a received frame.

BIT	BIT NAME	BIT DESCRIPTION
7..0	PsuedoCRCbyte0	The least significant byte of the PseudoCRC.
15..8	PsuedoCRCbyte1	The second lest significant byte of the PseudoCRC.
23..16	PsuedoCRCbyte2	The second most significant byte of the PseudoCRC.
31..24	PsuedoCRCbyte3	The most significant byte of the PseudoCRC.

MAGICSYNCSTREAM

Class.....Wake Event Data Structures, Magic Packet

Base Address .....Start of Magic Packet

Address Offset.....0x00

Access Mode.....Read only

Width .....48 bits

BIT	BIT NAME	BIT DESCRIPTION
47..0	MagicSyncStream	A stream of 6 bytes with the value 0xff indicates the start of the Magic-Sequence.

MAGICSEQUENCE

Class.....Wake Event Data Structures, Magic Packet

Base Address .....Start of Magic Packet

Address Offset.....0x06

Access Mode.....Read only

Width .....768 bits

BIT	BIT NAME	BIT DESCRIPTION
767..0	MagicSequence	A sequence of 96 bytes, consisting of 16 consecutive, identical 6 bytes sequences, where each 6 byte sequence equals the station address of the station receiving the Magic Packet.

I/O REGISTERS

The host interacts with the ST201 mainly through slave registers, which occupy 128 bytes in the host system's I/O space, memory space, or both. Generally, registers are referred to as "I/O registers", implying that the registers may in fact be mapped and accessed by the host system in memory space. I/O registers must be accessed with instructions that are no larger than the bit-width of that register. For instance, even though the FramesWithExcessiveDeferral, FramesLostRxErrors, and FramesWithDeferredXmission registers all appear in the same double word at offset 78, it is not legal to read all three registers with a single 32-bit I/O read instruction.

The ST201 I/0 register layout is show in Figure 11.

byte 3	byte 2	byte 1	byte 0	Offset
McstFramesRcvdOk	McstFramesXmtdOk	BcstFramesRcvdOk	BcstFramesXmtdOk	0x7c
FramesAbortXSColls	FramesWEXDeferral	FramesLostRxErrors	FramesWDeferedXmt	0x78
SingleColFrames	MultipleColFrames	LateCollisions	CarrierSenseErrors	0x74
FramesReceivedOk		FramesTransmittedOk		0x70
OctetsTransmittedOk(1)		OctetsTransmittedOk(0)		0x6c
OctetsReceivedOk(1)		OctetsReceivedOk(0)		0x68
HashTable(3)		HashTable(2)		0x64
HashTable(1)		HashTable(0)		0x60
	PhyCtrl	TxReleaseThresh	ReceiveMode	0x5c
MaxFrameSize		StationAddress(2)		0x58
StationAddress(1)		StationAddress(0)		0x54
MACCtrl(1)		MACCtrl(0)		0x50
IntStatus		IntEnable		0x4c
IntStatusAck		Countdown		0x48
TxFrameId	TxStatus	WakeEvent	ExpRomData	0x44
ExpRomAddr				0x40
RxEarlyThresh		TxSrartThresh		0x3c
FIFOCtrl				0x38
EepromCtrl		EepromData		0x34
AsicCtrl				0x30
				0x2c
				0x28
				0x24
				0x20
				0x1c
DebugCtrl				0x18
	RxDMAPollPeriod	RxDMAUrgentThresh	RxDMABurstThresh	0x14
RxDMAListPtr				0x10
RxDMAStatus				0x0c
	TxDMAPollPeriod	TxDMAUrgentThresh	TxDMABurstThresh	0x08
TxDMAListPtr				0x04
DMACtrl				0x00

FIGURE 11: ST201 I/O Register Layout

ASICCTRL

Class.....I/O Registers, Control and Status

Base Address .....IoBaseAddress register value

Address Offset.....0x30

Access Mode.....Read/Write

Width .....32 bits

AsicCtrl provides chip-specific, non-host-related settings. The contents of the least significant byte of AsicCtrl are read from EEPROM at reset.

BIT	BIT NAME	BIT DESCRIPTION
0	Reserved	Reserved for future use. Should be set to 0.
1	ExpRomSize	Specifies the size of the Expansion ROM installed on the adapter, as follows: 0 = 32 kB (default after reset) 1 = 64 kB
2	TxLargeEnable	This read/write bit, when set, enables transmission of frames that are larger than the TxFIFO. Since ST201's TxFIFO size is 2KB, this bit can be left clear (the reset default).
3	RxLargeEnable	This read/write bit, when set, enables reception of frames that are larger than the RxFIFO. Since ST201's RxFIFO size is 2KB, this bit can be left clear (the reset default).
4	ExpRomDisable	This bit, when set, disables accesses to the on-adapter Expansion ROM. This bit is included to allow bypassing the Expansion ROM without having to physically remove it from the board. When this bit is set, the ST201 responds to any read in its configured Expansion ROM space by returning 00000000h, and it ignores writes to the Expansion ROM. This bit resets to 0.
5	PhySpeed10	This read-only bit, when set, indicates the 10Mb/s operation is available from the PHY on the adapter. "0" indicates the PHY is not 10Mb/s capable.
6	PhySpeed100	This read-only bit, when set, indicates the 100Mb/s operation is available from the PHY on the adapter. "0" indicates the PHY is not 100Mb/s capable.
7	PhyMedia	This read-only bit indicates the media type that is available on the adapter. "0" indicates twisted-pair media, and "1" indicates fiber media. The combination of PhyMedia, PhySpeed100, and PhySpeed10 will determine the capability of the adapter. For example, [7,6,5] = 000: undefined 001: 10BASE-T PHY 010: 100BASE-T PHY 011: 10BASE-T and 100BASE-T dual-speed PHY 100: undefined 101: 10BASE-F PHY 110: 100BASE-F PHY 111: 10BASE-F and 100BASE-F dual-speed PHY

BIT	BIT NAME	BIT DESCRIPTION
10..8	ForcedConfig	These bits are used to place the ST201 into Forced Configuration mode. The bit values are latched in from ED[2..0] pins with a logic inversion at the end of RSTN or power on reset. 000: no forced configuration 001: forced configuration mode 1 010-111: reserved Note: When ForcedConfig[10] is set, the ST201 will use an alternate DeviceID and VendorID.
11	D3ResetDisable	This read/write bit, when set, indicates that the ST201 is configured for operation in an AMD-style (pre-ACPI) Wake-On-LAN environment. Specifically, when the ST201 is in the D3 power state, assertion of PCI RSTN will not reset the ST201. It is cleared upon reset.
12	Reserved	Reserved for future use. Should be set to 0.
13	SpeedupMode	This read/write bit is used for simulation only. When set, it indicates a speed-up mode to decrease simulation time. The bit value is latched in from ED5 pin with a logic inversion at the end of RSTN or power on reset.
14	LEDMode	This bit is used to control the LED outputs. When cleared (default after reset), the LED outputs are in mode 0; when set, the LED outputs are in mode 1. LEDPWRN: Mode 0: steady ON when power is applied, flashing when frames are being transmitted. Mode 1: ON all the time. LEDLNKN: Mode 0: steady ON when link is up, and flashing when frames are being received. Mode 1: steady ON when link is up, and flashing when frames are being transmitted or received. LEDDPLXN: Mode 0/1: steady ON when PHY is in full duplex mode, and flashing when collisions are being detected. LEDSPDN: Mode 0/1: steady ON when link speed is 100mb/s, and OFF when link speed is 10Mb/s.
15	RstOutPolarity	When set, RSTOUT is asserted high. When cleared, RSTOUT is asserted low.
16	GlobalReset	This is a self-clearing global reset bit for the entire ST201. This bit controls reset to various logic blocks depending on the values of the selection bits [24..19]. The ST201 should be re-initialized after a GlobalReset. The registers in the PCI configuration space are not reset by the GlobalReset; they are handled by the Power On Reset Test routine executed by the host.
17	RxReset	When set, will reset receive logic throughout the ST201, including network interface receive logic, RxFIFO control logic, and RxDMA Logic if the corresponding selection bits [21..19] are set. This bit is self-clearing. The RxReset should not be used after initialization except to recover from receive errors such as a RxFIFO under run.
18	TxReset	When set, will reset transmit logic throughout the ST201, including network interface transmit logic, TxFIFO control logic, and TxDMA Logic, if the corresponding selection bits [21..19] are set. This bit is self-clearing. The TxReset is required after a transmit under run error.

BIT	BIT NAME	BIT DESCRIPTION
19	DMA	When set, together with GlobalReset, RxReset, or TxReset bits, will reset RxDMA and TxDMA Logic, including: TxDMAListPtr, RxDMAListPtr, TxDMAComplete TxDMAInProg RxDMAComplete and RxEarlyEnable in DMACtrl and RxDMAStatus. When cleared, reset will not have action on the DMA Logic. This bit is self-clearing. Setting this bit has no meaning if the corresponding reset bits are not set.
20	FIFO	When set, together with GlobalReset, RxReset, or TxReset bits, will reset FIFO control logic, including TxStartThresh, TxReleaseThresh, and RxEarlyThresh. When cleared, reset will not have action on the DMA Logic. This bit is self-clearing. Setting this bit has no meaning if the corresponding reset bits are not set.
21	Network	When set, together with GlobalReset, RxReset, or TxReset bits, will reset network interface logic, including CSMA/CD MAC core, ReceiveMode, TxStatus, and the statistics registers. When cleared, reset will not have action on the network logic. This bit is self-clearing. Setting this bit has no meaning if the corresponding reset bits are not set.
22	Host	When set, together with GlobalReset bit, will reset host bus interface logic, including IntStatus, IntEnable, and Countdown. When cleared, reset will not have action on the host bus interface logic. This bit is self-clearing. Setting this bit has no meaning if the corresponding GlobalReset bit is not set.
23	AutoInit	When set, together with GlobalReset bit, will reset auto-initialize state machine logic and EEPROM data is reloaded. This bit is self-clearing. Setting this bit has no meaning if the corresponding GlobalReset bit is not set.
24	RstOut	When set, together with GlobalReset bit, will assert RSTOUT according to RstOutPolarity. When cleared, reset will not cause any action on RSTOUT. This bit is self-clearing. Setting this bit has no meaning if the corresponding GlobalReset bit is not set.
25	InterruptRequest	When set, the ST201 will assert IntRequested bit in the IntStatus register. This bit is self-clearing.
26	ResetBusy	When set, this bit indicates the reset is in progress. As the adapter's serial EEPROM may need to be read as part of the reset process, this operation can take as long as 1 ms to complete. The ResetBusy bit must be polled to be assured that the reset operation has completed.
31..27	Reserved	Reserved for future use. Should be set to 0.

DEBUGCTRL

Class.....I/O Registers, Control and Status

Base Address .....IoBaseAddress register value

Address Offset.....0x1a

Access Mode.....Read/Write

Width .....16 bits

DebugCtrl selects the functions of the GPIO pins. DebugCtrl is cleared by reset.

BIT	BIT NAME	BIT DESCRIPTION
0	GPIO0Ctrl	This bit controls the GPIO0 pin. When cleared, GPIO0 pin is an input. When set, GPIO0 pin is an output. This bit is cleared on reset.
1	GPIO1Ctrl	This bit controls the GPIO1 pins. When cleared, GPIO1 pin is an input. When set, GPIO1 pin is an output. This bit is cleared on reset.
2	GPIO0	When read, this bit shows the status values of the GPIO0 pin. When written into, this bit will drive the GPIO0 pin if GPIO0Ctrl is set, and GPIO0 pin functions as an output.
3	GPIO1	When read, this bit shows the status values of the GPIO1 pin. When written into, this bit will drive the GPIO1 pin if GPIO1Ctrl is set, and GPIO1 pin functions as an output.
15..4	Reserved	Reserved for future use. Should be set to 0.

HASHTABLE

Class.....I/O Registers, Control and Status

Base Address .....IoBaseAddress register value

Address Offset.....0x66, 0x64, 0x62, 0x60

Access Mode.....Read/Write

Width .....64 bits (accessible as 4, 16 bit words)

The host stores the 64-bit hash table in this register for selectively receiving multicast frames. Setting the ReceiveMulticastHash bit in ReceiveMode enables the filtering mechanism. The hash table is cleared upon reset, and must be properly programmed by the host.

BIT	BIT NAME	BIT DESCRIPTION
15..0	HashTableWord0	The least significant word of the hash table, corresponding to address 0x60.
31..16	HashTableWord1	The second least significant word of the hash table, corresponding to address 0x62.
47..32	HashTableWord2	The second most significant word of the hash table, corresponding to address 0x64.
63..48	HashTableWord3	The most significant word of the hash table, corresponding to address 0x66.

The ST201 applies a cyclic-redundancy-check (the same CRC used to calculate the frame data FCS) to the destination address all incoming multicast frames (with multicast bit set). The low-order 6 bits of the CRC result are used as an addressing index into the hash table. The MSB of HashTable(3) is the most significant, and the LSB of HashTable(0) is the least significant bit, addressed by the 6-bit index. If the HashTable bit addressed by the index is set, the frame is accepted by the ST201 and transferred to higher layers. If the addressed hash table bit is cleared, the frame is discarded.

MACCTRL

Class.....I/O Registers, Control and Status

Base Address .....IoBaseAddress register value

Address Offset.....0x50

Access Mode.....Read/Write

Width .....32 bits

This register provides for setting of MAC-specific parameters. It is cleared upon reset.

BIT	BIT NAME	BIT DESCRIPTION
1..0	IFSSelect	This field is used to select the size of Inter-Frame Spacing (IFS). By programming a larger number of bit times for the IFS, the ST201 will become less “aggressive” on the network and may defer more than normal. The performance of the ST201 may decrease as the IFS value is increased from the standard value. This, however, will prevent the ST201 from “capturing” the network. 00: no IFS extension, IFS = 96 BT (802.3 standard value). 01: IFS extended by 32 BT, IFS = 128 BT. 10: IFS extended by 128 BT, IFS = 224 BT. 11: IFS extended by 448 BT, IFS = 544 BT.
4..2	Reserved	Reserved for future use. Should be set to 0.
5	FullDuplexEnable	Setting this bit configures the ST201 to function in a full duplex manner. Specifically, it disables transmitter deference to receive traffic, allowing simultaneous receive and transmit traffic. It has the side-effect of disabling CarrierSenseErrors statistics collection, since full duplex operation requires carrier sense to be masked to the transmitter. TxReset and RxReset bits in AsicCtrl must be set after changing the value of this bit.
6	RcvLargeFrames	This bit determines the frame size at which the OversizedFrame error is generated for receive frames. When RcvLargeFrames is cleared, minimum OversizedFrame size is 1514 bytes. When RcvLargeFrames is set, minimum OversizedFrame size is 4491 bytes. (This value was the maximum FDDI frame size of 4500 bytes, subtracting bytes for fields that have no Ethernet equivalent.) The frame size at which an OversizedFrame error will be flagged includes the destination and source addresses, the type/length field, and the FCS field.
7	Reserved	Reserved for future use. Should be set to 0.
8	FlowControlEnable	Flow control enable. When it is cleared (default), the ST201 treats all incoming frames as data frames. When it is set, Flow control is enabled and the ST201 will act upon incoming flow control PAUSE frame. Note that FlowControlEnable should not be set unless FullDuplexEnable is also set.

BIT	BIT NAME	BIT DESCRIPTION
9	RcvFCS	This bit is set by the host if it is desired for the receive frame's FCS to be passed to the host as part of the data in the RxFIFO. The state of RcvFCS does not affect the ST201's checking of the frame's FCS and its posting of FCS error status. RcvFCS is cleared by a system reset. To avoid confusing the RxFIFO logic, the value of RcvFCS should only be changed when the receiver is disabled and the RxFIFO is empty.
10	FIFOLoopback	Setting this bit forces data loopback from the TxFIFO directly into the RxFIFO. When using FIFO Loopback mode, it is the software's responsibility to ensure that the proper interframe gap is inserted between frames, to avoid losing data in the receive path. To do this, the software must not load more than one transmit frame into the TxFIFO at a time. TxReset and RxReset bits in AsicCtrl must be set after changing the value of this bit.
11	MACLoopback	Setting this bit will cause the ST201 to loop back transmissions at the output of the media access controller. TxReset and RxReset bits in AsicCtrl must be set after changing the value of this bit.
15..12	Reserved	Reserved for future use. Should be set to 0.
16	CollisionDetect	This read-only bit provides a real-time indication of the state of the COL signal within ST201.
17	CarrierSense	This read-only bit provides a real-time indication of the state of the CRS signal within ST201.
18	TxInProg	A real-time indication that a frame is being transmitted. This bit is used by drivers during under run recovery to delay issuing a TxReset.
19	TxError	If a TxUnderrun occurs, this bit is set, indicating that the transmitter needs to be reset with the TxReset.
20	Reserved	Reserved for future use. Should be set to 0.
21	StatisticsEnable	When set, ST201 will collect statistics with the various statistics counters and registers. This bit is self-clearing.
22	StatisticsDisable	When set, ST201 will stop collection of statistics with the statistics registers. The values in the statistics registers will remain unchanged. This bit is self-clearing.
23	StatisticsEnabled	This read-only status bit is set by ST201 to indicate that statistics collection is enabled.
24	TxEnable	When set, the transmitter logic is enabled. This bit is self-clearing.
25	TxDisable	When set, the transmitter logic is disabled. This bit is self-clearing.
26	TxEnabled	This read-only status bit is set by ST201 to indicate that transmitter is enabled.
27	RxEnable	When set, the receiver logic is enabled. This bit is self-clearing.
28	RxDisable	When set, the receiver logic is disabled. This bit is self-clearing.
29	RxEnabled	This read-only status bit is set by ST201 to indicate that receiver is enabled.

BIT	BIT NAME	BIT DESCRIPTION
30	Paused	This read-only status bit is set by ST201 to indicate that a PAUSE MAC Control frame had been received and halted the transmit MAC for the duration of the pause_time. It is cleared when the MAC can resume transmission.
31	Reserved	Reserved for future use. Should be set to 0.

The loopback modes available to a host system when using the ST201 are shown in Table 3.

Loopback Mode	FIFOLoopback	MACLoopback	FullDuplexEnable
FIFO Loopback	1	0	x
MAC Loopback	0	1	x
External Loopback (MII) or True “On-wire” External	0	0	1

TABLE 3: ST201 Loopback Modes

External loopback type is controlled by the Mll PHY device. The host system must enable a loopback mode within MII PHY device using the MII Management Interface. For the true “on-the-wire” loopback mode, use a loopback plug (connector), clear all of the loopback bits, set the FullDuplexEnable bit in the MACCtrl register, and enable the full duplex mode within the MII PHY device.

MAXFRAMESIZE

Class.....I/O Registers, Control and Status

Base Address .....IoBaseAddress register value

Address Offset.....0x5a

Access Mode.....Read/Write

Width .....16 bits

Sets the maximum frame size for received frames.

BIT	BIT NAME	BIT DESCRIPTION
15..0	MaxFrameSize	Received frames with sizes equal to or larger than the value in Max-FrameSize will be flagged as oversize by RxOversizedFrame bit in RxDMAStatus. MaxFrameSize defaults to 1514 upon reset. Upon RxReset, MaxFrameSize is automatically loaded with 1514 or 4491 depending upon the value of RcvLargeFrames in MACCtrl.

RECEIVEMODE

Class.....I/O Registers, Control and Status

Base Address .....IoBaseAddress register value

Address Offset.....0x5c

Access Mode.....Read/Write

Width .....8 bits

Each bit in ReceiveMode, when set, enables reception of a different type of frame. ReceiveMode is cleared upon reset.

BIT	BIT NAME	BIT DESCRIPTION
0	ReceiveUnicast	Setting this bit enables the ST201 to receive unicast frames that match the 48-bit StationAddress of the ST201.
1	ReceiveMulticast	Setting this bit causes the ST201 to receive all multicast frames, including broadcast.
2	ReceiveBroadcast	Setting this bit causes the ST201 to receive all broadcast frames.
3	ReceiveAllFrames	Setting this bit causes the ST201 to receive all frames promiscuously.
4	ReceiveMulticast-Hash	Setting this bit enables the ST201 to receive frames that pass the hash filtering mechanism.
5	ReceiveIPMulticast	Setting this bit enables the ST201 to receive all multicast IP datagrams, which are mapped into Ethernet multicast frames with destination address of 01:00:5e:xx:xx:xx as defined in RFC 1112 and RFC 1700. The first 3 bytes require exact match, and the last 3 bytes are ignored.
7..6	Reserved	Reserved for future use. Should be set to 0.

STATIONADDRESS

Class.....I/O Registers, Control and Status

Base Address .....IoBaseAddress register value

Address Offset.....0x47

Access Mode.....Read/Write

Width .....8 bits

StationAddress is used to define the individual destination address that the ST201 will respond to when receiving frames. Network addresses are generally specified in the form of 01:23:45:67:89:ab, where the bytes are received left to right, and the bits within each byte are received right to left (lsb to msb). The actual transmitted and received bits are in the order of 10000000 11000100 10100010 11100110 10010001 11010101.

BIT	BIT NAME	BIT DESCRIPTION
15..0	StationAddress Word0	The least significant word of the station, corresponding to address 0x54.
31..16	StationAddress Word1	The second least significant word of the station, corresponding to address 0x56.
47..32	StationAddress Word2	The most significant word of the station, corresponding to address 0x58.

The address comparison logic will compare the first 16 received destination address bits against Station-Address(0), the second 16 received destination address bits against StationAddress(1), and the third 16 received destination address bits against StationAddress(2). The value set in the StationAddress register is not inserted into the source address field of frames transmitted by the ST201. The source address field for every frame must be specified by the host system as part of the frame data contents.

TXFRAMEID

Class.....I/O Registers, Control and Status

Base Address .....IoBaseAddress register value

Address Offset.....0x5c

Access Mode.....Read

Width .....8 bits

TxFrameId contains the frame ID for the currently transmitting or most recently transmitted frame.

BIT	BIT NAME	BIT DESCRIPTION
7..0	TxFrameId	This register contains the value from FrameId sub-field within the frame's TFD, TransmitFrameControl field.

Host systems can use TxFrameId register during transmit error recovery by scanning through the TFD's in the TxDMAList, searching for a match between the TxFrameId register value and a FrameId value in the TFD.

TXSTATUS

Class.....I/O Registers, Control and Status

Base Address .....IoBaseAddress register value

Address Offset.....0x46

Access Mode.....Read (write to advance queue)

Width .....8 bits

The TxStatus register returns the status of frame transmission or transmission attempts. TxStatus actually implements a queue of up to 31 transmit status bytes. An I/O write of an arbitrary value to TxStatus will advance the queue to the next transmit status byte.

BIT	BIT NAME	BIT DESCRIPTION
0	Reserved	Reserved for future use. Should be set to 0.
1	TxReleaseError	Indicates that a transmit release error occurred, meaning that the frame transmission experienced a collision after the front of the frame had already been released to the TxFIFO free space. Refer to TxReleaseThresh register for complete description.
2	TxStatusOverflow	When set, indicates that the TxStatus stack is full and as a result the transmitter has been disabled. Writing TxStatus clears this bit, but the transmitter must be re-enabled with the TxEnable before transmissions may resume.
3	MaxCollisions	When set, the frame was not successfully transmitted due to encountering 16 collisions. TxEnable must be set to recover from this condition. The frame is discarded from the TxFIFO, so driver should resubmit the frame for transmission.
4	TxUnderrun	This bit indicates that the frame experienced an under run during the transmit process because the host was unable to supply the frame data fast enough to keep up with the network data rate. An under run will halt the transmitter and the TxFIFO. The TxReset and TxEnable bits must be set prior to re-starting any frame.
5	Reserved	Reserved for future use. Should be set to 0.
6	TxIndicateReqd	This bit is asserted if the TxIndicate bit was set when the 32-bit TransmitFrameControl was written to the ST201 for the frame.
7	TxComplete	If this bit is cleared, then the remainder of the status bits are undefined. If the host chooses to poll this register while waiting for a frame transmission to complete, then this bit is used to determine that a frame transmission attempt has either experienced an error, or has completed successfully with the TxIndicate bit set in the TransmitFrameControl.

WAKEEVENT

Class.....I/O Registers, Control and Status

Base Address .....IoBaseAddress register value

Address Offset.....0x45

Access Mode.....Read/Write

Width .....8 bits

WakeEvent contains enable bits to control which types of events can generate a wake event to the host system. It also contains status bits indicating the specific events that have occurred.

BIT	BIT NAME	BIT DESCRIPTION
0	WakePktEnable	Setting this read/write bit enables the ST201 to generate wake events via a PCI interrupt due to Wake Packet reception. PmeEn must be set in the PowerMgmtCtrl register in order for WakePktEnable to be recognized. WakePktEnable has no effect in power mode D0.
1	MagicPktEnable	Setting this read/write bit enables the ST201 to generate wake events via a PCI interrupt due to Magic Packet reception. MagicPktEnable is set after a ST201 reset. PmeEn must be set in the PowerMgmtCtrl register in order for MagicPktEnable to be recognized. MagicPktEnable has no effect in power mode D0.
2	LinkEventEnable	Setting this read/write bit enables the ST201 to generate wake events via a PCI interrupt due to a change in link status (cable connect or disconnect). PmeEn must be set in the PowerMgmtCtrl register in order for LinkEventEnable to be recognized. LinkEventEnable has no effect in power mode D0.
3	WakePolarity	Setting this read/write bit will cause the Wake pin to be asserted in the HIGH state (default after RESET). MagicPktEnable is set after a ST201 reset.
4	WakePktEvent	Indicates that a wake packet (which meets the reception criteria set by driver) has been received. WakePktEvent is masked by WakePktEnable, and must be set for WakePktEvent to operate. WakePktEvent is cleared when the WakeEvent register is read.
5	MagicPktEvent	Indicates that a magic packet has been received. MagicPktEvent is masked by MagicPktEnable, and must be set for MagicPktEvent to operate. MagicPktEvent is cleared when the WakeEvent register is read.
6	LinkEvent	Indicates that a link status event has occurred. LinkEvent is masked by LinkEventEnable, and must be set for LinkEvent to operate. LinkEvent is cleared when the WakeEvent register is read.
7	WakeOnLanEnable	Setting this read/write bit puts the ST201 in the WakeOnLan mode regardless of the power management register settings in the configuration space.

FIFOCTRL

Class.....I/O Registers, FIFO Control

Base Address .....IoBaseAddress register value

Address Offset.....0x3a

Access Mode.....Read/Write

Width .....16 bits

The bits in this register provide various control and indications of TxFIFO and RxFIFO diagnostic.

BIT	BIT NAME	BIT DESCRIPTION
0	RAMTestMode	When set, the FIFO RAM is in the test mode. This bit is cleared after reset.
8..1	Reserved	Reserved for future use. Should be set to 0.
9	RxOverrunFrame	This read/write bit determines how the ST201 handles receive overrun frames. The default is zero, which causes the ST201 to discard all overrun frames. Setting this bit causes the ST201 to keep and make visible all overrun frames that have been made visible to the host, so that they may be inspected for diagnostic purposes.
10	Reserved	Reserved for future use. Should be set to 0.
11	RxFIFOFull	This read-only bit is set when the RxFIFO is full. This bit does not in itself indicate an overrun condition. However, if more data is received while this bit is set, an overrun will occur. This bit is informational in nature only. This bit is cleared as soon as the RxFIFO is no longer full.
13..12	Reserved	Reserved for future use. Should be set to 0.
14	Transmitting	Transmitting is read-only and set by ST201 whenever the MAC is transmitting or waiting to transmit (deferring).
15	Receiving	This read-only bit is set whenever the ST201 is receiving a frame into the RxFIFO. No particular action is expected on the part of the host based on the state of this bit.

RXEARLYTHRESH

Class.....I/O Registers, FIFO Control

Base Address .....IoBaseAddress register value

Address Offset.....0x3e

Access Mode.....Read/Write

Width .....16 bits

The value stored in this register defines the number of bytes of the top of the frame that must be received before a RxEarly interrupt will occur. The first byte of the destination address is considered to be byte 1. The value in RxEarlyThresh may also determine how many bytes of a frame must be received before RxDMA transfers for the frame are allowed to begin. If RxEarlyEnable in DMACtrl is set, a frame is not eligible to start RxDMA until RxEarlyThresh bytes have been received. RxEarlyThresh resets to the value 1ffch, which disables the threshold mechanism.

BIT	BIT NAME	BIT DESCRIPTION
12..0	RxEarlyThresh	The number of bytes which must be present in the RxFIFO before a RxEarly interrupt is asserted. The minimum value is 0x08, any value smaller than this will be interpreted as 0x08. Values greater than 0x08 will be interpreted using a resolution of 4 bytes (i.e. 0x08, 0x0c, 0x0f, etc.)
15..13	Unused	These bits are ignored.

TXRELEASETHRESH

Class.....I/O Registers, FIFO Control

Base Address .....IoBaseAddress register value

Address Offset.....0x5d

Access Mode.....Read/Write

Width .....8 bits

The value in TxReleaseThresh determines how much data of a frame must be transmitted before the TxFIFO space can be released for use by another frame. Once the number of bytes equal to the value in TxReleaseThresh have been transmitted, that number of bytes are discarded from the TxFIFO. Thereafter, bytes are discarded as they are transmitted to the network. TxReleaseThresh resets to 8, i.e., a release threshold of 128 bytes. A value of 255 in TxReleaseThresh disables the release mechanism and TxFIFO frame space is not released until the entire frame is transmitted. A TxReleaseError is signaled in TxStatus when a frame experiences a collision after its release threshold has been crossed, preventing MAC from retry. When a release error occurs, the transmitter is disabled, and the frame's ID or sequence number is visible in TxFrameId.

BIT	BIT NAME	BIT DESCRIPTION
7..0	TxReleaseThresh	The number of 16 byte words which must be transmitted before the space in the TxFIFO occupied by the transmitted data can be released. To avoid excessive release errors due to in-window collisions, value less than 4 should not be written into TxReleaseThresh.

TXSTARTTHRESH

Class.....I/O Registers, FIFO Control

Base Address .....IoBaseAddress register value

Address Offset.....0x3c

Access Mode.....Read/Write

Width .....16 bits

The value in TxStartThresh is used to control when frames are transmitted. Transmission of a frame begins when the number of bytes for the frame transferred into the TxFIFO is greater than the value in TxStartThresh. If TxStartThresh is set too low, the TxFIFO may experience under run due to the DMA data transfers not able to keep up with the wire data rate. Host systems should use under run indications as a hint to increase the TxStartThresh value. This register resets to 1ffch, which disables the threshold mechanism.

BIT	BIT NAME	BIT DESCRIPTION
12..0	TxStartThresh	The number of bytes which must be present in the TxFIFO before frame transmission begins. Values will be interpreted using a resolution of 4 bytes (i.e. 0x00, 0x04, 0x08, 0x0c, 0x0f, etc.)
15..13	Unused	These bits are ignored.

COUNTDOWN

Class.....I/O Registers, Interrupt

Base Address .....IoBaseAddress register value

Address Offset.....0x48

Access Mode.....Read/Write

Width .....16 bits

Countdown is a programmable down-counter that will generate an interrupt upon its expiration. If the CountdownIntEnable bit in DMACtrl is set, the IntRequested interrupt will be generated when Countdown counts through zero. Countdown has two modes of operation that is selected by the CountdownMode bit in DMACtrl.When CountdownMode is cleared, Countdown is loaded by the host software with an initial countdown value, then decrements at a rate determined by the CountdownSpeed bit in DMACtrl. When CountdownSpeed is cleared, the count rate is once every 3.2 us. When CountdownSpeed is set, the count rate is once every 320 ns. When Countdown reaches zero, it continues to count down, wrapping to 0xffff. When CountdownMode is set, Countdown begins counting only when TxDMAComplete in IntStatus becomes set. Countdown is cleared by reset.

BIT	BIT NAME	BIT DESCRIPTION
15..0	Countdown	Value of current state of Countdown timer.

INTENABLE

Class.....I/O Registers, Interrupt

Base Address .....IoBaseAddress register value

Address Offset.....0x4c

Access Mode.....Read/Write

Width .....16 bits

Enables individual interrupts as specified in the IntStatus register. Setting a bit in IntEnable will allow the specific source to generate an interrupt on the PCI bus. IntEnable is cleared upon reset. IntEnable is also cleared by a read of IntStatusAck.

BIT	BIT NAME	BIT DESCRIPTION
0	Unused	This bit will be ignored.
1	EnHostError	Enables the HostError interrupt.
2	EnTxComplete	Enables the TxComplete interrupt.
3	EnMACControl-Frame	Enables the MACControlFrame interrupt.
4	EnRxComplete	Enables the RxComplete interrupt.
5	EnRxEarly	Enables the RxEarly interrupt.
6	EnInRequested	Enables the InRequested interrupt.
7	EnUpdateStats	Enables the UpdateStats interrupt.
8	EnLinkEvent	Enables the LinkEvent interrupt.
9	EnTxDMACom-plete	Enables the TxDMAComplete interrupt.
10	EnRxDMACom-plete	Enables the RxDMAComplete interrupt.
15..11	Unused	These bits bill be ignored.

INTSTATUS

Class.....I/O Registers, Interrupt

Base Address .....IoBaseAddress register value

Address Offset.....0x4e

Access Mode.....Read/Write

Width .....16 bits

IntStatus register indicates the source of interrupts and indications on the ST201. Bits 1 through 10 are the interrupt-causing sources for the ST201. These bits can be individually disabled as interrupt sources using the IntEnable register. The host can acknowledge the interrupt by writing a “1” into the indication bit(s), which will cause ST201 to clear the interrupt indication. IntStatus is cleared by reset.

BIT	BIT NAME	BIT DESCRIPTION
0	InterruptStatus	Asserted when the ST201 is driving the bus interrupt signal. It is a logical OR of all the interrupt-causing bits after they have been filtered through the IntEnable register.
1	HostError	This bit is set when a catastrophic error related to the bus interface occurs. The errors which set HostError are: PCI target abort and PCI master abort. HostError is cleared by GlobalReset/DMA bits set.
2	TxComplete	This bit is asserted when a frame whose TxIndicate bit is set has been successfully transmitted or for any frame that experiences a transmission error. This interrupt is acknowledged by writing to TxStatus to advance the status queue.
3	MACControlFrame	This bit is set when a MAC Control frame has been received by the ST201. MACControlFrame is acknowledged by writing a 1 to this bit.
4	RxComplete	This bit is set when one or more entire frames have been received into the RxFIFO. This bit is automatically acknowledged by the RxDMA Logic as it transfers frames. Drivers should disable this interrupt and mask this bit when reading IntStatus.
5	RxEarly	This bit is set when the number of bytes of the top frame that have been received is greater than the value of RxEarlyThresh. When the top frame has been completely received by the ST201, RxEarly will be negated and RxComplete will assert. RxEarly is acknowledged by writing a 1 to this bit.
6	IntRequested	This bit is set when the host requested interrupt by setting InterruptRequest bit or by the expiration of the Countdown. It is acknowledged by writing a 1 to this bit.
7	UpdateStats	This bit indicates that one or more of the statistics counters is nearing overflow condition (typically half of its maximum value). A driver should respond to an UpdateStats interrupt by reading all of the statistics, thereby acknowledging and clearing UpdateStats bit.
8	LinkEvent	This bit indicates transition of link status of the PHY. This bit can be acknowledged by writing a 1 into this bit.

BIT	BIT NAME	BIT DESCRIPTION
9	TxDMAComplete	This bit indicates that a frame TxDMA has completed, and the TFD in question had the TxDMAIndicate bit in its TFC set. This bit can be acknowledged by writing a 1 to this bit. The host should examine the TxDMAListPtr to determine which frame(s) have been transferred by TxDMA. Those frames in the TxDMAList before the current TxDMAListPtr have already been transferred by TxDMA. If the TxDMAListPtr is zero, then all the frames are transmitted.
10	RxDMAComplete	This bit indicates that a frame RxDMA has completed. This bit can be acknowledged by writing a 1 to this bit.
15..11	Reserved	Reserved for future use. Should be set to 0.

INTSTATUSACK

Class.....I/O Registers, Interrupt

Base Address .....IoBaseAddress register value

Address Offset.....0x4a

Access Mode.....Read only

Width .....16 bits

IntStatusAck is another version of the IntStatus register, having the same bit definition as IntStatus, but providing additional functionality to reduce the number of I/O operations required to perform common tasks related to interrupt handling. In addition to returning the IntStatus value for the specified interrupt, when read IntStatusAck also acknowledges the TxDMAComplete, RxDMAComplete, RxEarly, IntRequested, MACControlFrame, and LinkEvent bits within the IntStatus register (if they are set), and clears the IntEnable register (preventing subsequent events from generating an interrupt).

BIT	BIT NAME	BIT DESCRIPTION
0	InterruptStatus	Asserted when the ST201 is driving the bus interrupt signal. It is a logical OR of all the interrupt-causing bits after they have been filtered through the IntEnable register.
1	HostError	This bit is set when a catastrophic error related to the bus interface occurs. The errors which set HostError are: PCI target abort and PCI master abort. HostError is cleared by GlobalReset/DMA bits set.
2	TxComplete	This bit is asserted when a frame whose TxIndicate bit is set has been successfully transmitted or for any frame that experiences a transmission error. This interrupt is acknowledged by writing to TxStatus to advance the status queue.
3	MACControlFrame	This bit is set when a MAC Control frame has been received by the ST201. MACControlFrame is acknowledged by writing a 1 to this bit.
4	RxComplete	This bit is set when one or more entire frames have been received into the RxFIFO. This bit is automatically acknowledged by the RxDMA Logic as it transfers frames. Drivers should disable this interrupt and mask this bit when reading IntStatus.
5	RxEarly	This bit is set when the number of bytes of the top frame that have been received is greater than the value of RxEarlyThresh. When the top frame has been completely received by the ST201, RxEarly will be negated and RxComplete will assert. RxEarly is acknowledged by writing a 1 to this bit.
6	IntRequested	This bit is set when the host requested interrupt by setting InterruptRequest bit or by the expiration of the Countdown. It is acknowledged by writing a 1 to this bit.
7	UpdateStats	This bit indicates that one or more of the statistics counters is nearing overflow condition (typically half of its maximum value). A driver should respond to an UpdateStats interrupt by reading all of the statistics, thereby acknowledging and clearing UpdateStats bit.
8	LinkEvent	This bit indicates transition of link status of the PHY. This bit can be acknowledged by writing a 1 into this bit.

BIT	BIT NAME	BIT DESCRIPTION
9	TxDMAComplete	This bit indicates that a frame TxDMA has completed, and the TFD in question had the TxDMAIndicate bit in its TFC set. This bit can be acknowledged by writing a 1 to this bit. The host should examine the TxDMAListPtr to determine which frame(s) have been transferred by TxDMA. Those frames in the TxDMAList before the current TxDMAListPtr have already been transferred by TxDMA. If the TxDMAListPtr is zero, then all the frames are transmitted.
10	RxDMAComplete	This bit indicates that a frame RxDMA has completed. This bit can be acknowledged by writing a 1 to this bit.
15..11	Reserved	Reserved for future use. Should be set to 0.

DMACTRL

Class.....I/O Registers, DMA

Base Address .....IoBaseAddress register value

Address Offset.....0x00

Access Mode.....Read/Write

Width .....32 bits

DMACtrl controls some of the bus master functions in the RxDMA and TxDMA engines, and contains status bits. DMACtrl is cleared by a reset.

BIT	BIT NAME	BIT DESCRIPTION
0	RxDMAHalted	This read-only bit is set whenever RxDMA is halted by setting the RxDMAHalt bit or an implicit halt due to fetching a RFD with RxDMAComplete in ReceiveFrameStatus already set. Cleared by setting the RxDMAResume bit.
1	TxDMACmplReq	This read-only bit is set to the value from the TxDMAIndicate field in the TFC of the current TFD.
2	TxDMAHalted	This read-only bit is set whenever TxDMA is halted by setting the TxDMAHalt bit. Cleared by setting the TxDMAResume bit.
3	RxDMAComplete	This read-only bit is the same as RxDMAComplete in IntStatus. This bit is different from the RxDMAComplete bit in RxDMAStatus, which is a real time indication of frame RxDMA completion and is cleared when a new frame RxDMA begins. This bit is latched once a frame RxDMA completion has occurred. This bit is cleared by acknowledgment to the RxDMAComplete bit in the IntStatus register.
4	TxDMAComplete	This read-only bit is the same as TxDMAComplete in IntStatus. This bit is cleared by acknowledgment to the TxDMAComplete bit in the IntStatus register.
7..5	Reserved	Reserved for future use. Should be set to 0.
8	RxDMAHalt	Whenever this bit is set, the RxDMA is halted. This bit is self-clearing and writing a 0 into this bit is ignored.
9	RxDMAResume	Whenever this bit is set, the RxDMA is resumed. This bit is self-clearing and writing a 0 into this bit is ignored.
10	TxDMAHalt	Whenever this bit is set, the TxDMA is halted. This bit is self-clearing and writing a 0 into this bit is ignored.
11	TxDMAResume	Whenever this bit is set, the TxDMA is resumed. This bit is self-clearing and writing a 0 into this bit is ignored.
13..12	Reserved	Reserved for future use. Should be set to 0.
14	TxDMAInProg	This read-only bit indicates that a TxDMA operation is in progress. This bit is primarily used by drivers in an under run recovery routine since the driver needs waits for this bit to be cleared before issuing the TxReset to clear the under run condition. Before checking TxDMAInProg, issue TxDMAHalt to ensure that TxDMAInProg is not set as a result of the ST201 being in a polling mode.

BIT	BIT NAME	BIT DESCRIPTION
15	DMAHaltBusy	This read-only bit indicates that a DMA Halt operation (TxDMAHalt or RxDMAHalt) is in progress and the drivers should wait for this bit to be cleared before performing other actions.
16	Reserved	Reserved for future use. Should be set to 0.
17	RxEarlyEnable	This read/write bit determines when the ST201 may start RxDMA a receive frame. By default (cleared), RxDMA qualify for bus-master arbitration when the frame becomes visible, normally at 60 bytes unless a RxEarlyThresh threshold smaller than that has been set. When set to one, RxDMA won't start until the RxEarlyThresh threshold has been crossed (or the frame completes, whichever is first).
18	CountdownSpeed	This read/write bit sets the speed at which Countdown counts. When CountdownSpeed is cleared, the count rate is once every 3.2 us (i.e. 4 byte times at 10 Mbps). When CountdownSpeed is set, the count rate is once every 320 ns (i.e. 4 byte times at 100 Mbps). By setting appropriate CountdownSpeed for the wire speed, conversions can be made between byte times and counter values using simple shift operations.
19	CountdownMode	This read/write bit controls the operating mode of the Countdown register. With this bit cleared, Countdown begins its down counting operation as soon as a non-zero value is written to it. With this bit set, Countdown will not begin counting down until TxDMAComplete (in IntStatus) is set. See the Countdown register definition for more information on the Countdown modes.
20	MWIDisable	Setting this read/write bit prevents the bus master logic from using the Memory Write Invalidate (MWI) PCI command.
21	Reserved	Reserved for future use. Should be set to 0.
22	RxDMAOverrun-Frame	This read/write bit, when clear (the default), causes the RxDMA engine to discard receive overrun frames without transferring them to system memory. When this bit is set, the RxDMA engine keeps and transfers overrun frames.
23	CountdownIntEnable	This read-only bit specifies whether expiration of Countdown can generate interrupts. If CountdownIntEnable is clear, Countdown expiration will not set IntRequested; if it is set, expiration of Countdown will set IntRequested. CountdownIntEnable is managed completely by the hardware. This bit is cleared automatically by the act of setting IntRequested or when a zero value is written into Countdown. This bit is set implicitly when a non-zero value is written into Countdown by the host. This allows the host to write a non-zero value to Countdown and an interrupt will be generated in a corresponding amount of time. By writing a zero value to Countdown the host can suppress interrupts.
29..24	Reserved	Reserved for future use. Should be set to 0.
30	TargetAbort	This read-only bit is set when the ST201 experiences a target abort sequence when operating as a bus master. This bit indicates a fatal error, and must be cleared before further TxDMA or RxDMA operation can proceed. This bit is cleared by the GlobalReset/DMA bit.

BIT	BIT NAME	BIT DESCRIPTION
31	MasterAbort	This read-only bit is set when the ST201 experiences a master abort sequence when operating as a bus master. This bit indicates a fatal error, and must be cleared before further TxDMA or RxDMA operation can proceed. This bit is cleared by the GlobalReset/DMA bit.

RXDMABURSTTHRESH

Class.....I/O Registers, DMA

Base Address .....IoBaseAddress register value

Address Offset.....0x14

Access Mode.....Read/Write

Width .....8 bits

RxDMABurstThresh sets the threshold when the ST201 makes RxDMA bus master requests, based upon the number of used bytes in the RxFIFO, in units of 32 bytes. When the used space exceeds the threshold, the ST201 may make a RxDMA request on the PCI bus. However, if the used space exceeds the current RxDMAFragLen, ST201 will make RxDMA bus request regardless of whether the used space exceeds the RxDMABurstThresh or not. RxDMABurstThresh may be overridden by the urgent request mechanism. See the PCI Bus Master Operation section for information about the relationship between RxDMABurst-Thresh and RxDMAUrgentThresh. A value of zero is invalid. RxDMABurstThresh defaults to 8, a threshold of 256 bytes.

BIT	BIT NAME	BIT DESCRIPTION
7..0	RxDMABurst-Thresh	The number of 32 byte words which must be present in the RxFIFO prior to the assertion of a RxDMA bus master request.

RXDMALISTPTR

Class.....I/O Registers, DMA

Base Address .....IoBaseAddress register value

Address Offset.....0x10

Access Mode.....Read/Write

Width .....32 bits

RxDMAListPtr holds the physical address of the current RxDMA Frame Descriptor in the RxDMAList. A value of zero in RxDMAListPtr indicates that no more RFDs are available to accept receive frames. RxDMAListPtr only points to addresses on 8-byte boundaries, so RFDs must be aligned on 8-byte physical address boundaries. RxDMAListPtr is cleared by reset. RxDMAListPtr may be written directly by the host system to point the ST201 to the head of a newly created RxDMAList. RxDMAListPtr is also updated by the ST201 as it processes RFDs in the RxDMAList. As the ST201 finishes processing a RFD, it loads RxDMAListPtr with the value from RxDMANextPtr to allow it to move on to the next RFD. If the ST201 loads a value of zero from the current RFD, the RxDMA engine enters the idle state, waiting for a non-zero value to be written to RxDMAListPtr. To avoid access conflicts between the ST201 and the host system, the host system must issue a RxDMAHalt before writing to RxDMAListPtr.

BIT	BIT NAME	BIT DESCRIPTION
31..0	RxDMAListPtr	Physical address, on a 8-byte boundary, of the current RFD in the RxDMAList.

RXDMASTATUS

Class.....I/O Registers, DMA

Base Address .....IoBaseAddress register value

Address Offset.....0x0c

Access Mode.....Read only

Width .....32 bits

RxDMAStatus shows the status of various operations in the RxDMA Logic. Host systems should read this register only while the RxDMA engine is in the RxDMAHalt state. Otherwise the ST201 may change RFDs between accesses to this register. The format of this register is identical to that of the ReceiveFrameStatus field written into each RFD, since the content of this register is written into the RFD upon RxDMA frame completion with the exception of the ImpliedBufferEnable bit, which is not implemented in this register. RxDMAStatus is cleared by a reset.

BIT	BIT NAME	BIT DESCRIPTION
12..0	RxDMAFrameLen	During frame RxDMA, RxDMAFrameLen gives a real-time indication of the number of bytes transferred by RxDMA for the frame. RxDMAFrameLen is cleared when the ST201 fetches a new RxDMAListPtr, and counts up in steps no larger than a bus master burst. When the frame has been completely transferred by RxDMA, RxDMAFrameLen indicates the true frame length, except in the case where the frame is larger than the number of bytes specified in the RxDMA fragments. In this case, the RxDMAOverflow bit will be set.
13	Reserved	Reserved for future use. Should be set to 0.
14	RxFrameError	Indicates that an error occurred in the receipt of the frame. The driver should examine bits 16 through 20 to determine the type of error(s). This bit is undefined until RxDMAComplete bit is set.
15	RxDMAComplete	Indicates that the frame transfer by RxDMA is complete. Unless a RxDMA halt is in effect this bit would normally only remain set momentarily (too short for the software to read it) since the hardware will then fetch the next RFD.
16	RxFIFOOverrun	Indicates that the hardware was unable to remove data from the RxFIFO quickly enough (most likely because the software failed to free a RFD quickly enough, or kept the ST201 in the RxDMAHalt state for too long). Bytes will be missing from the frame at one or more locations in the frame (unpredictable). This bit is undefined until RxDMAComplete bit is set.
17	RxRuntFrame	Indicates that the frame was a runt (less than 60 bytes). Normally such frames are not transferred by RxDMA unless RxEarlyThresh is set to a value less than the actual size of the runt frame, and the RxEarlyEnable of MacCtrl register must be set. This bit is undefined until RxDMAComplete bit is set.
18	RxAlignmentError	Indicates that the frame had an alignment error (bad FCS and dribble bits). This bit is undefined until RxDMAComplete bit is set.
19	RxFCSError	Indicates a FCS checksum error on the frame data. This bit is undefined until RxDMAComplete bit is set.

BIT	BIT NAME	BIT DESCRIPTION
20	RxOversizedFrame	Indicates the frame size was equal to or greater than the value set in the MaxFrameSize register. This bit is undefined until RxDMAComplete bit is set.
22..21	Reserved	Reserved for future use. Should be set to 0.
23	DribbleBits	Indicates that the frame had accompanying dribble bits. This bit is informational only, and does not indicate a frame error.
24	RxDMAOverflow	Indicates that the RFD had insufficient buffer space for the frame data and there were still data left to be transferred by RxDMA when the ST201 ran out of fragment space. The ST201 will transfer what it can into the buffers provided, discard the remainder of the frame and set this bit.
31..25	Reserved	Reserved for future use. Should be set to 0.

RXDMAPOLLPERIOD

Class.....I/O Registers, DMA

Base Address .....IoBaseAddress register value

Address Offset.....0x16

Access Mode.....Read/Write

Width .....8 bits

The value in RxDMAPollPeriod determines the rate at which the current RFD is polled, looking for RxDMA-Complete in ReceiveFrameStatus in RFD to be cleared. Polling is disabled when RxDMAPollPeriod is cleared. RxDMAPollPeriod is cleared by reset. The value in RxDMAPollPeriod represents multiple of 320 ns time intervals. The maximum value is 127 (or 40.64 us).

BIT	BIT NAME	BIT DESCRIPTION
6..0	RxDMAPollPeriod	The number of 320ns intervals between polls of the RxDMAComplete bit in the ReceiveFrameStatus field of the current RFD.
7	Unused	This bit is ignored.

RXDMAURGENTTHRESH

Class.....I/O Registers, DMA

Base Address .....IoBaseAddress register value

Address Offset.....0x15

Access Mode.....Read/Write

Width .....8 bits

The value in RxDMAUrgentThresh sets a threshold at which the RxDMA engine will make a urgent bus master request. A urgent RxDMA request will have priority over all other requests on the ST201. The urgent bus request is made when the free space in the RxFIFO falls below the value in RxDMAUrgent-Thresh. A RxDMA urgent request is not subject to the RxDMABurstThresh constraint. When the RxFIFO is close to overrun, burst efficiency is sacrificed in favor of requesting the bus as quickly as possible. The value in RxDMAUrgentThresh represents free space in the RxFIFO in terms of 32-byte portions. RxDMAU-rgentThresh resets to 4, or a threshold of 128 bytes.

BIT	BIT NAME	BIT DESCRIPTION
4..0	RxDMAUrgent-Thresh	The minimum number of 32-byte words which must be available in the RxFIFO to avoid assertion of a RxDMA Urgent Request.
7..5	Unused	These bits are ignored.

TXDMABURSTTHRESH

Class.....I/O Registers, DMA

Base Address .....IoBaseAddress register value

Address Offset.....0x08

Access Mode.....Read/Write

Width .....8 bits

TxDMABurstThresh determines the threshold for when the ST201 makes TxDMA bus master requests, based upon the available space in the TxFIFO. The value in TxDMABurstThresh represents free space in the TxFIFO in multiples of 32 bytes. When the free space exceeds the threshold, the ST201 may make a TxDMA request. However, if the free space exceeds the current TxDMAFragLen, ST201 will make TxDMA bus request regardless of whether the free space exceeds the TxDMABurstThresh or not. TxDMABurst-Thresh may be overridden by the TxDMAUrgentThresh mechanism. See the PCI Bus Master Operation section for information about the relationship between TxDMABurstThresh and TxDMAUrgentThresh. A value of zero is invalid. TxDMABurstThresh defaults to 8, a threshold of 256 bytes.

BIT	BIT NAME	BIT DESCRIPTION
4..0	TxDMABurst-Thresh	The number of 32-byte words which must be available in the TxFIFO prior to assertion of a TxDMA Burst Request.
7..5	Unused	These bits are ignored.

TXDMALISTPTR

Class.....I/O Registers, DMA

Base Address .....IoBaseAddress register value

Address Offset.....0x04

Access Mode.....Read/Write

Width .....32 bits

TxDMAListPtr holds the physical address of the current TxDMA Frame Descriptor in the TxDMAList. A value of zero in TxDMAListPtr is interpreted by the ST201 to mean that no more frames remain to be transferred by TxDMA. TxDMAListPtr can only point to addresses on 8-byte boundaries, so TFD's must be aligned on 8-byte boundaries. TxDMAListPtr is cleared by reset. TxDMAListPtr may be written directly by the host system to point the ST201 at the head of a newly created TxDMAList. Writes to TxDMAListPtr are ignored while the current value in TxDMAListPtr is non-zero. To avoid access conflicts between the ST201 and the host system, the host system must issue a TxDMAHalt before writing to TxDMAListPtr (unless the host system has specific knowledge that TxDMAListPtr contains zero). TxDMAListPtr is also updated by the ST201 as it processes TFD's in the TxDMAList. As the ST201 finishes processing a TFD, it fetches the value from TxDMANextPtr. If it is zero, the TxDMA engine becomes idle. Also, if TxDMA polling is enabled (TxDMAPollPeriod is non-zero), the old value in TxDMAListPtr is preserved. If the value fetched from TxDMANextPtr is non-zero, then the value is stored temporarily in the ST201, and the ST201 inspects the TFD at that location. The temporary value is loaded into TxDMAListPtr, advancing the ST201 to the new TFD. There are two ways the TxDMA engine can leave the idle state. First, the driver can write a nonzero value directly to TxDMAListPtr. Second, if polling is enabled, then the TxDMA engine will leave the idle state when a non-zero value is finally fetched from TxDMANextPtr. Reading TxDMAListPtr while the TxDMA engine is polling has the following side effects:

1. Any pending decision to advance to the next TFD is cancelled.
2. The TxDMA engine fetches the TxDMANextPtr in the current (completed) TFD immediately, rather than waiting for the full TxDMAPollPeriod interval. (It is assumed that the host system will only read TxDMAListPtr when it is in the process of inserting a TFD at the list head, so it will have written a new value into TxDMANextPtr to hook up the inserted TFD).

BIT	BIT NAME	BIT DESCRIPTION
31..0	TxDMAListPtr	Physical address, on a 8-byte boundary, of the current TFD in the TxDMAList.

TXDMAPOLLPERIOD

Class.....I/O Registers, DMA

Base Address .....IoBaseAddress register value

Address Offset.....0x0a

Access Mode.....Read/Write

Width .....8 bits

The value in TxDMAPollPeriod determines the interval at which the current TFD is polled. When a zero TxDMANextPtr is fetched from the current TFD, TxDMANextPtr is polled to determine when a new TFD is ready to be processed. Polling is disabled when TxDMAPollPeriod is cleared. TxDMAPollPeriod is cleared by reset. The value in TxDMAPollPeriod represents a multiple of 320 ns time intervals. The maximum value is 127 (or 40.64 us).

BIT	BIT NAME	BIT DESCRIPTION
6..0	TxDMAPollPeriod	The number of 320ns intervals between polls of the TxDMANextPtr field of the current TFD.
7	Unused	This bit is ignored.

TXDMAURGENTTHRESH

Class.....I/O Registers, DMA

Base Address .....IoBaseAddress register value

Address Offset.....0x09

Access Mode.....Read/Write

Width .....8 bits

When the number of used bytes in the TxFIFO falls below the value in the TxDMAUrgentThresh, the TxDMA Logic will make an urgent bus master request. An urgent TxDMA request will have priority over the RxDMA, unless it is also making an urgent request. A TxDMA urgent request is not subject to the TxDMABurstThresh constraint. The relaxation of the TxDMABurstThresh constraint for this condition is because the TxFIFO is close to under run, and burst efficiency is sacrificed to avoid FIFO under run. The value in TxDMAUrgentThresh represents data in the TxFIFO in multiples of 32 bytes. TxDMAUrgent-Thresh resets to 4, or a threshold of 128 bytes.

BIT	BIT NAME	BIT DESCRIPTION
5..0	TxDMAUrgent-Thresh	The minimum number of 32-byte words which must be occupied in the TxFIFO to avoid assertion of a TxDMA Urgent Request.
7..6	Unused	These bits are ignored.

EEPROMCTRL

Class.....I/O Registers, External Interface Control

Base Address .....IoBaseAddress register value

Address Offset.....0x36

Access Mode.....Read/Write

Width .....16 bits

EepromCtrl provides the host with a method for issuing commands to the ST201's serial EEPROM controller. Individual 16-bit word locations within the EEPROM may be written, read or erased. Also, the EEPROM's WriteEnable, WriteDisable, EraseAll and WriteAll commands can be issued. Two-bit opcodes and 8-bit addresses are written to this register to cause the ST201 to carry out the desired EEPROM command. If data is to written to the EEPROM, the 16-bit data word must be written to EepromData by the host prior to issuing the associated write command. Similarly, if data is to be read from the EEPROM, the read data will be available via EepromData register. The EEPROM is a particularly slow device. It is important that the host wait until the EepromBusy bit is false before issuing a command to EepromCtrl. EepromCtrl defaults to 0000h upon reset.

BIT	BIT NAME	BIT DESCRIPTION
7..0	EepromAddress	These eight read/write bits identify one of the 256 sixteen-bit words to be the target for the ReadRegister, WriteRegister and EraseRegister commands. Bits 7 and 6 are further defined to identify an individual command among the following group of four sub-commands: sub-opcodesub-command 00WriteDisable 01WriteAII 10EraseAII 11WriteEnable The definition of bits 7 and 6 are valid when the EepromOpcode in bits 9 and 8 equals 00.
9..8	EepromOpcode	These two read/write bits specify one of three individual commands and a single group of four sub-commands. The following table defines the opcodes: EepromOpcode command 00 Write Enable/Disable & Write/Erase All sub-commands 01 WriteRegister 10 ReadRegister 11 EraseRegister
14..10	Reserved	Reserved for future use. Should be set to 0.
15	EepromBusy	This read-only bit is asserted during the execution of EEPROM commands. Further commands should not be issued to EepromCtrl nor should data be read from EepromData while this bit is true.

EEPROMDATA

Class.....I/O Registers, External Interface Control

Base Address .....IoBaseAddress register value

Address Offset.....0x34

Access Mode.....Read/Write

Width .....16 bits

EepromData is a 16-bit data register for use with the adapter's serial EEPROM. Data from the EEPROM can be read by the host from EepromData register after EepromBusy is cleared. Data to be written to the EEPROM is written to EepromData prior to issuing the write command to EepromCtrl. EepromData is cleared after a system reset.

BIT	BIT NAME	BIT DESCRIPTION
15..0	EepromData	Data read from, or to be written to, the external EEPROM.

EXPROMADDR

Class.....I/O Registers, External Interface Control

Base Address .....IoBaseAddress register value

Address Offset.....0x40

Access Mode.....Read/Write

Width .....32 bits

ExpRomAddr holds the address to be used for direct I/O accesses of the Expansion ROM through the ExpRomData port. To access a byte in the Expansion ROM, write the address of the byte to be accessed into ExpRomAddr. Then issue either a read or a write to ExpRomData. For reads, the ROM value will be returned by the read instruction. For writes, the new value will be programmed into the ROM upon completion of the write instruction.

BIT	BIT NAME	BIT DESCRIPTION
15..0	ExpRomAddr	Address used for accessing expansion ROM.
31..16	Reserved	Reserved for future use. Should be set to 0.

EXPROMDATA

Class.....I/O Registers, External Interface Control

Base Address .....IoBaseAddress register value

Address Offset.....0x44

Access Mode.....Read/Write

Width .....8 bits

ExpRomData is the data port for performing direct I/O byte-wide accesses of the Expansion ROM. A read of ExpRomData returns the ROM byte value from the location specified by ExpRomAddr. A write to ExpRomData causes the write data to be programmed into the ROM location specified by ExpRomAddr. Note: The Atmel EPROM devices supported by ST201 must be programmed in 64-byte pages. Refer to the Atmel Flash Memory Device data book for information on programming EPROMs.

BIT	BIT NAME	BIT DESCRIPTION
15..0	ExpRomData	Data read from, or to be written to, expansion ROM.

PHYCTRL

Class.....I/O Registers, External Interface Control

Base Address .....IoBaseAddress register value

Address Offset.....0x5e

Access Mode.....Read/Write

Width .....8 bits

This register contains control bits for the MII Management Interface. The MII Management Interface is used to access registers in an MII PHY device. The Management Interface is a two-wire serial interface connecting ST201 to any MII-compliant PHY devices residing on the adapter. The host system operates the Management Interface by writing and reading bit patterns to the PhyCtrl register which correspond to the physical waveforms required on the interface signals. For more information on the Management Interface signal protocols, refer to the Media Independent Interface standard of IEEE 802.3u Specification.

BIT	BIT NAME	BIT DESCRIPTION
0	MgmtClk	The MII Management Clock. This bit drives directly the management clock to the PHY device(s).
1	MgmtData	The MII Management Data bit. When the MgmtDir bit (below) is set, the value written to this bit is driven onto the MDIO signal. When MgmtDir is cleared, data being driven by the PMD can be read from this bit.
2	MgmtDir	The MII data direction control bit. Setting this bit causes ST201 to drive MDIO with the data bit written into MgmtData.
3	DisableClk25	This bit is set to tri-state the CLK25 pin. DisableClk25 is cleared upon reset.
4	PhyDuplexPolarity	Clearing this read/write bit will cause the PHYDPLXN input pin to be active low. The default value for this bit upon reset is cleared.
5	PhyDuplexStatus	This read-only bit provides a real-time indication of the duplex status of the PHY. This bit is set for full duplex operation.
6	PhySpeedStatus	This read-only bit provides a real-time indication of the speed status of the PHY. This bit is set for 100Mb/s operation.
7	PhyLinkStatus	This read-only bit provides a real-time indication of the twisted-pair transceiver link. This bit is set for operational link (link status up).

STATISTICS

Reading a statistic register will clear it. The statistics gathering must be enabled by setting the StatisticsEnable bit in MACCtrl for the statistics registers to count events.

BROADCASTFRAMESRECEIVEDOK

Class.....I/O Registers, Statistics

Base Address .....IoBaseAddress register value

Address Offset.....0x7d

Access Mode.....Read (also clears register)/Write

Width .....8 bits

BIT	BIT NAME	BIT DESCRIPTION
7..0	BroadcastFrames-ReceivedOk	This statistic counts the number of frames that are successfully received and are directed to the broadcast group address. This does not include frames received with frame-too-long, FCS, length or alignment errors, or frames lost due to internal MAC error. This is a 8-bit counter and will wrap around to zero after reaching ffh. An UpdateStats interrupt will occur, if enabled, after the counter counts through c0h.

BROADCASTFRAMESTRANSMITTEDOK

Class.....I/O Registers, Statistics

Base Address .....IoBaseAddress register value

Address Offset.....0x7c

Access Mode.....Read (also clears register)/Write

Width .....8 bits

BIT	BIT NAME	BIT DESCRIPTION
7..0	Broadcast-FramesTransmitte-dOk	This statistic counts the number of frames that are successfully transmitted to the broadcast address. Frames transmitted to multicast addresses are excluded from this statistic. This is a 8-bit counter and will wrap around to zero after reaching ffh. An UpdateStats interrupt will occur, if enabled, after the counter counts through c0h.

CARRIERSENSEERRORS

Class.....I/O Registers, Statistics

Base Address .....IoBaseAddress register value

Address Offset.....0x74

Access Mode.....Read (also clears register)/Write

Width .....8 bits

BIT	BIT NAME	BIT DESCRIPTION
3..0	CarrierSenseErrors	This statistic register counts the number of times that carrier_sense was not asserted or was de-asserted during the transmission of a frame without collision. Carrier sense is not monitored for the purpose of this statistic until after the preamble and start-of-frame delimiter. This is a 4-bit counter that will stick at 0fh. An UpdateStats interrupt will occur, if enabled, after the counter has counted through 0ch.
7..4	Reserved	Reserved for future use. Should be set to 0.

FRAMESABORTEDDUETOXSCOLLS

Class.....I/O Registers, Statistics

Base Address .....IoBaseAddress register value

Address Offset.....0x7b

Access Mode.....Read (also clears register)/Write

Width .....8 bits

BIT	BIT NAME	BIT DESCRIPTION
7..0	FramesAbortedDueToXSColls	This statistic counts the number of frames that, due to excessive collisions, are not transmitted successfully. This is an 8-bit counter and will wrap around to zero after reaching ffh. An UpdateStats indication will occur after the counter has counted through c0h.

FRAMESLOSTRXERRORS

Class.....I/O Registers, Statistics

Base Address .....IoBaseAddress register value

Address Offset.....0x79

Access Mode.....Read (also clears register)/Write

Width .....8 bits

BIT	BIT NAME	BIT DESCRIPTION
7..0	FramesLostRxErrors	This statistic counts the number of frames that should have been received (the destination address matched the filter criteria) but experienced a RxFIFO overrun error due to there not being enough space to hold the frame. This statistic only includes overruns that become apparent to the driver, and does not count frames that are completely ignored due to the RxFIFO being full at the start of frame reception. This is an 8-bit counter and will wrap around to zero after reaching ffh. An UpdateStats indication will occur after the counter has counted through c0h.

FRAMESRECEIVEDOK

Class.....I/O Registers, Statistics

Base Address .....IoBaseAddress register value

Address Offset.....0x72

Access Mode.....Read (also clears register)/Write

Width .....16 bits

BIT	BIT NAME	BIT DESCRIPTION
7..0	FramesReceive-dOk	This statistic counts the number of frames that are successfully received. This does not include frames with frame-too-long, FCS, length or alignment errors, or frames lost due to internal MAC error. Error frames will have Overrun, RuntFrame, AlignmentError, FCSError, or OversizedFrame is set in RxDMAStatus. This is a 16-bit counter and will wrap around to zero after reaching ffffh. An UpdateStats interrupt will occur, if enabled, after the counter counts through c000h.

FRAMESTRANSMITTEDOK

Class.....I/O Registers, Statistics

Base Address .....IoBaseAddress register value

Address Offset.....0x70

Access Mode.....Read (also clears register)/Write

Width .....16 bits

BIT	BIT NAME	BIT DESCRIPTION
7..0	FramesTransmitte-dOk	This statistic counts the number of frames that are successfully transmitted. This is a 16-bit counter and will wrap around to zero after reaching ffffh. An UpdateStats interrupt will occur, if enabled, after the counter counts through c000h.

FRAMESWITHDEFERREDXMISSION

Class.....I/O Registers, Statistics

Base Address .....IoBaseAddress register value

Address Offset.....0x78

Access Mode.....Read (also clears register)/Write

Width .....8 bits

BIT	BIT NAME	BIT DESCRIPTION
7..0	FramesWithDeferredXmission	This statistic counts the number of frames that must delay its first attempt of transmission because the medium was busy. Frames involved in any collisions are not counted by this statistic. This is an 8-bit counter and will wrap around to zero after reaching ffh. An UpdateStats interrupt will occur after the counter has counted through c0h.

FRAMESWITHEXCESSIVEDEFERAL

Class.....I/O Registers, Statistics

Base Address .....IoBaseAddress register value

Address Offset.....0x7a

Access Mode.....Read (also clears register)/Write

Width .....8 bits

BIT	BIT NAME	BIT DESCRIPTION
7..0	FramesWithExces-siveDeferal	This statistic counts the number of frames that deferred for an excessive period of time (exceeding the defer limit). This statistic is only incremented once per LLC frame. This is an 8-bit counter and will wrap around to zero after reaching ffh. An UpdateStats interrupt will occur after the counter has counted through c0h.

LATECOLLISIONS

Class.....I/O Registers, Statistics

Base Address .....IoBaseAddress register value

Address Offset.....0x75

Access Mode.....Read (also clears register)/Write

Width .....8 bits

BIT	BIT NAME	BIT DESCRIPTION
7..0	LateCollisions	This statistic counts the number of times that a collision has been detected later than 512 BT into the transmitted frame. A late collision is counted both as a Collision and a LateCollision. This is an 8-bit counter and will wrap around to zero after reaching ffh. An UpdateStats interrupt will occur, if enabled, after the counter counts through c0h.

MULTICASTFRAMESRECEIVEDOK

Class.....I/O Registers, Statistics

Base Address .....IoBaseAddress register value

Address Offset.....0x7f

Access Mode.....Read (also clears register)/Write

Width .....8 bits

BIT	BIT NAME	BIT DESCRIPTION
7..0	MulticastFrames-ReceivedOk	This statistic counts the number of frames that are successfully received and are directed to an active non-broadcast group address. This does not include frames received with frame-too-long, FCS, length or alignment errors, or frames lost due to internal MAC error. This is a 8-bit counter and will wrap around to zero after reaching ffh. An UpdateStats interrupt will occur, if enabled, after the counter counts through c0h.

MULTICASTFRAMESTRANSMITTEDOK

Class.....I/O Registers, Statistics

Base Address .....IoBaseAddress register value

Address Offset.....0x7e

Access Mode.....Read (also clears register)/Write

Width .....8 bits

BIT	BIT NAME	BIT DESCRIPTION
7..0	Multicast-FramesTransmitte-dOk	This statistic counts the number of frames that are successfully transmitted to a group destination address other than broadcast. This is a 8-bit counter and will wrap around to zero after reaching ffh. An UpdateStats interrupt will occur, if enabled, after the counter counts through c0h.

MULTIPLECOLLISIONFRAMES

Class.....I/O Registers, Statistics

Base Address .....IoBaseAddress register value

Address Offset.....0x76

Access Mode.....Read (also clears register)/Write

Width .....8 bits

BIT	BIT NAME	BIT DESCRIPTION
7..0	MultipleCollision-Frames	This statistic counts the number of frames that are involved in more than one collision and are subsequently transmitted successfully. This is a 8-bit counter and will wrap around to zero after reaching ffh. An UpdateStats interrupt will occur, if enabled, when the counter has counted through c0h.

OCTETSRECEIVEDOK

Class.....I/O Registers, Statistics

Base Address .....IoBaseAddress register value

Address Offset.....0x68

Access Mode.....Read (also clears register)/Write

Width .....32 bits

BIT	BIT NAME	BIT DESCRIPTION
19..0	OctetsReceivedOk	This statistic counts the total number of frame header, data and padding octets in frames that are successfully received. For the purposes of this statistic, a successfully received frame is one that is completely moved into the RxFIFO. This is a 20-bit counter and will wrap around to zero after reaching fffffh. An UpdateStats interrupt will occur after the counter has counted through c0000h. This statistic must be read as two 16-bit quantity with the lower word first. Upon reading the lower word, the upper word value is latched in and the lower value is cleared. The upper word is also cleared when read.
31..20	Reserved	Reserved for future use. Should be set to 0.

OCTETSTRANSMITTEDOK

Class.....I/O Registers, Statistics

Base Address .....IoBaseAddress register value

Address Offset.....0x6c

Access Mode.....Read (also clears register)/Write

Width .....32 bits

BIT	BIT NAME	BIT DESCRIPTION
19..0	OctetsTransmitte-dOk	This statistic counts the total number of frame header, data and padding octets in frames that are successfully transmitted. This is a 20-bit counter and will wrap around to zero after reaching fffffh. An UpdateStats interrupt will occur after the counter has counted through c0000h. This statistic must be read as two 16-bit quantity with the lower word first. Upon reading the lower word, the upper word value is latched in and the lower value is cleared. The upper word is also cleared when read.
31..20	Reserved	Reserved for future use. Should be set to 0.

SINGLECOLLISIONFRAMES

Class.....I/O Registers, Statistics

Base Address .....IoBaseAddress register value

Address Offset.....0x77

Access Mode.....Read (also clears register)/Write

Width .....8 bits

BIT	BIT NAME	BIT DESCRIPTION
7..0	SingleCollision-Frames	This statistic counts the number of frames that are involved in a single collision, and are subsequently transmitted successfully. This is a 8-bit counter and will wrap around to zero after reaching ffh. An UpdateStats interrupt will occur after the counter has counted through c0h.

PCI CONFIGURATION REGISTERS

PCI based systems use a slot-specific block of configuration registers to perform configuration of devices on the PCI bus. The configuration registers are accessed with PCI Configuration Cycles. The PCI bus supports two types of Configuration Cycles. Type 0 cycles are used to configure devices on the local PCI bus. Type 1 cycles are used to pass a configuration request to a PCI bus at a different hierarchical level. PCI Configuration Cycles are directed at one out of eight possible PCI logical functions within a single physical PCI device. A ST201 based PCI bus master device responds only to Type 0 Configuration Cycles, directed at function 0. Type 1 cycles, and Type 0 cycles directed at functions other than 0, are ignored by the ST201.

Each PCI bus device is required to decode 256 bytes of configuration registers. Of these, the first 64 bytes are pre-defined by the PCI Specification. The remaining registers may be used as needed for PCI device-specific configuration registers. In PCI Configuration Cycles, the host system provides a slot-specific decode signal (IDSEL) which informs the PCI device that a configuration cycle is in progress. The PCI device responds by asserting DEVSELN, and decoding the specific configuration register from the address bus and the byte enable signals. See the PCI Expansion ROM specification for information on generating configuration cycles from driver software.

Figure 12 shows the PCI configuration registers implemented by ST201. All locations marked “Reserved”, and all of the locations within the 256-byte configuration space that are not shown in the table, are not implemented and return zero when read.

byte 3	byte 2	byte 1	byte 0	Offset
				0xe0..
Reserved				0x60
Reserved				0x5c
Reserved				0x58
Reserved		PowerMgmtCtrl		0x54
PowerMgmtCap		NextItemPtr	CapId	0x50
Reserved				0x4c
Reserved				0x48
Reserved				0x44
Reserved				0x40
MaxLat	MinGnt	InterruptPin	InterruptLine	0x3c
Reserved				0x38
Reserved			CapPtr	0x34
ExpRomBaseAddress				0x30
SubsystemId		SubsystemVendorId		0x2c
Reserved				0x28
Reserved				0x24
Reserved				0x20
Reserved				0x1c
Reserved				0x18
MemBaseAddress				0x14
IoBaseAddress				0x10
Reserved	HeaderType	LatencyTimer	CacheLineSize	0x0c
ClassCode			RevisionId	0x08
ConfigStatus		ConfigCommand		0x04
DeviceId		VendorId		0x00

FIGURE 12: ST201 PCI Register Layout

CACHELINESIZE

Class.....PCI Configuration Registers, Configuration

Base Address **.....PCI device configuration header start**

Address Offset.....0x0c

Access Mode.....Read/Write

Width .....8 bits

BIT	BIT NAME	BIT DESCRIPTION
7..0	CacheLineSize	The system BIOS writes the system's cache line size into this register. The adapter uses this to optimize PCI bus master operation (choosing the best memory command, etc.). The value in CacheLineSize represents the number of dwords in a cache. CacheLineSize only supports powers of two from 4 to 64 (giving a range of 16 to 256 bytes). An unsupported value is treated the same as zero.

CAPPTR

Class.....PCI Configuration Registers, Configuration

Base Address **.....PCI device configuration header start**

Address Offset.....0x34

Access Mode.....Read Only

Width .....8 bits

BIT	BIT NAME	BIT DESCRIPTION
7..0	CapPtr	This is a hard-coded value pointing to the beginning of a chain of registers that describe enhanced functions. The CapPtr register returns 50h, which points to the power management registers.

CLASSCODE

Class.....PCI Configuration Registers, Configuration

Base Address **.....PCI device configuration header start**

Address Offset.....0x09

Access Mode.....Read Only

Width .....24 bits

BIT	BIT NAME	BIT DESCRIPTION
23..0	ClassCode	This register identifies the general function of the PCI device. The ST201 returns 020000h, indicating Ethernet network controller.

CONFIGCOMMAND

Class.....PCI Configuration Registers, Configuration

Base Address **.....PCI device configuration header start**

Address Offset.....0x04

Access Mode.....Read/Write

Width .....16 bits

This register provides control over the adapter's ability to generate and respond to PCI cycles. When a zero is written to this register, the adapter is logically disconnected from the PCI bus, except for configuration cycles.

BIT	BIT NAME	BIT DESCRIPTION
0	IoSpace	Setting this bit allows the adapter to respond to I/O space accesses (if the adapter is in the D0 power state).
1	MemorySpace	Setting this bit along with AddressDecodeEnable in ExpRomBaseAddress allows the adapter to decode accesses to its Expansion ROM, if one is installed, and if the adapter is in the D0 power state.
2	BusMaster	Setting this bit allows adapters with bus master capability to initiate bus master cycles (if the adapter is in the D0 power state).
3	Reserved	Reserved for future use. Should be set to 0.
4	MWIEnable	Memory Write and Invalidate Enable. Setting this bit allows the adapter to generate the MWI command.
5	Reserved	Reserved for future use. Should be set to 0.
6	ParityErrorResponse	This bit controls how the adapter responds to parity errors. Setting this bit causes the adapter to take its normal action upon detecting a parity error. Clearing this bit causes the adapter to ignore parity errors. This bit is cleared upon system reset.
7	Reserved	Reserved for future use. Should be set to 0.
8	SERREnable	This bit is the enable bit for the SERRN pin driver. A value of zero disables the SERRN driver.
15..9	Reserved	Reserved for future use. Should be set to 0.

CONFIGSTATUS

Class.....PCI Configuration Registers, Configuration

Base Address **.....PCI device configuration header start**

Address Offset.....0x06

Access Mode.....Read/Write

Width .....16 bits

This register is used to record status information for PCI bus events. Read/write bits in the register can only be reset, not set, by writing to this register. Bits are reset by writing a one to that bit position.

BIT	BIT NAME	BIT DESCRIPTION
3..0	Reserved	Reserved for future use. Should be set to 0.
4	Capabilities	This read-only bit is always set, indicating the existence of a list of extended capabilities registers. The CapPtr register points to the start of the list.
6..5	Reserved	Reserved for future use. Should be set to 0.
7	FastBackToBack	This read-only bit, when set, indicates that the adapter, as a Target, supports fast back-to-back transactions as defined by the criteria in the section 3.4.2 of the PCI specification.
8	DataParityReported	The adapter sets this bit when, as a master, it detects the PERRN signal asserted, and the ParityErrorResponse bit is set in the ConfigCommand register.
10..9	DevselTiming	This read-only field is used to encode the slowest time with which the adapter asserts the DEVSELN signal. ST201-based adapters return 01b, indicating that they support “medium” speed DEVSELN assertion.
11	SignaledTargetAbort	The adapter sets this bit when it terminates a bus transaction with target-abort.
12	ReceivedTargetAbort	The adapter sets this bit when, operating as a bus master, its bus transaction is terminated with target-abort.
13	ReceivedMasterAbort	The adapter sets this bit when, operating as a bus master, its bus transaction is terminated with master-abort.
14	SignaledSystemError	This bit is set whenever the adapter asserts SERRN.
15	DetectedParityError	The adapter sets this bit when it detects a parity error, regardless of whether parity error handling is enabled.

DEVICEID

Class.....PCI Configuration Registers, Configuration

Base Address **.....PCI device configuration header start**

Address Offset.....0x02

Access Mode.....Read Only

Width .....16 bits

BIT	BIT NAME	BIT DESCRIPTION
15..0	DeviceId	This register contains the 16-bit device ID for the ST201. It is hard-wired to 0201h.

EXPROMBASEADDRESS

Class.....PCI Configuration Registers, Configuration

Base Address **.....PCI device configuration header start**

Address Offset.....0x30

Access Mode.....Read/Write

Width .....32 bits

This read/write register allows the system to define the base address for the adapter's Expansion ROM.

BIT	BIT NAME	BIT DESCRIPTION
0	AddressDecodeEnable	When this bit is cleared, the adapter's Expansion ROM is disabled. Setting this bit causes the adapter to respond to accesses in its configured expansion ROM space, if MemorySpace in the ConfigCommand register is also set.
14..1	Reserved	Reserved for future use. Should be set to 0.
31..15	RomBaseAddress	The system programs the expansion ROM base address into this field. The access to bit [15] depends on Expansion ROM size setting in AsicCtrl register. When ExpRomSize is 0 (32KB ExpRom), bit [15] is written as part of the RomBaseAddress. When ExpRomSize is 1 (64KB ExpRom), bit [15] ignores any write operation.

HEADERTYPE

Class.....PCI Configuration Registers, Configuration

Base Address **.....PCI device configuration header start**

Address Offset.....0x0e

Access Mode.....Read Only

Width .....8 bits

BIT	BIT NAME	BIT DESCRIPTION
7..0	HeaderType	This register is hard-wired to 00h, identifying the ST201 as a single-function PCI and specifies the configuration register layout.

INTERRUPTLINE

Class.....PCI Configuration Registers, Configuration

Base Address **.....PCI device configuration header start**

Address Offset.....0x3c

Access Mode.....Read/Write

Width .....8 bits

BIT	BIT NAME	BIT DESCRIPTION
7..0	InterruptLine	This register is written by the system to communicate to the device driver which interrupt level is being used for the device. This allows the driver to use the appropriate interrupt vector for its ISR. For 80x86 systems, the value in InterruptLine correspond to the IRQ numbers (0 through 15) of the standard dual 8259 configuration, and the value 255 correspond to “disabled”.

INTERRUPTPIN

Class.....PCI Configuration Registers, Configuration

Base Address **.....PCI device configuration header start**

Address Offset.....0x3d

Access Mode.....Read Only

Width .....8 bits

BIT	BIT NAME	BIT DESCRIPTION
7..0	InterruptPin	This register indicates which PCI interrupt pin the adapter will use. ST201-based adapters always use INTAN, so 01h is returned in InterruptPin.

IOBASEADDRESS

Class.....PCI Configuration Registers, Configuration

Base Address **.....PCI device configuration header start**

Address Offset.....0x10

Access Mode.....Read/Write

Width .....32 bits

The host uses this register to define the I/O base address for the adapter. PCI system requires that I/O base addresses be set as if the system used 32-bit I/O addressing. The upper 25 bits of the register are read/write accessible, indicating that the ST201 requires 128 bytes of I/O space in the system I/O map.

BIT	BIT NAME	BIT DESCRIPTION
0	IoBaseAddrInd	A value of 1 indicates this register holds the I/O base address for the ST201.
6..1	Reserved	Reserved for future use. Should be set to 0.
31..7	IoBaseAddress	The system programs the I/O base address into this field. Since the ST201 uses 128 bytes of I/O space, 25 bits are required to completely specify the I/O base address.

LATENCYTIMER

Class.....PCI Configuration Registers, Configuration

Base Address **.....PCI device configuration header start**

Address Offset.....0x0d

Access Mode.....Read/Write

Width .....8 bits

This register specifies, in units of PCI bus clocks, the value of the latency timer for bus master operations. The host system writes a value into LatencyTimer, which determines how long the ST201 may hold the bus in the presence of other bus requestors. Whenever the ST201 asserts FRAMEN, the latency timer is started. When the timer count expires, the ST201 must relinquish the bus as soon as its GNTN signal has been negated. The granularity of the timer is 8 bus clocks.

BIT	BIT NAME	BIT DESCRIPTION
2..0	Reserved	Reserved for future use. Should be set to 0.
7..3	LatencyTimer	Indicates, in increments of 8 bus clocks, the length of time which the ST201 may hold the PCI bus in the presence of other bus requestors.

MAXLAT

Class.....PCI Configuration Registers, Configuration

Base Address **.....PCI device configuration header start**

Address Offset.....0x3f

Access Mode.....Read Only

Width .....8 bits

BIT	BIT NAME	BIT DESCRIPTION
7..0	MaxLat	MaxLat specifies, in 250 ns increments, how often the ST201 requires bus access while operating as a bus master. The value for MaxLat is stored in the ConfigParm word in EEPROM. Assumes the PCI system allows 64-byte maximum bursts with full duplex operation, the ST201-based 100 Mbps systems return the value 10 in this field, implying a latency tolerance of 2.5us, according to the calculation: 64 bytes/(12.5 MB/s x 2) = 2.56us The result is rounded to 2.5 us, which is 250 ns * 10

MEMBASEADDRESS

Class.....PCI Configuration Registers, Configuration

Base Address **.....PCI device configuration header start**

Address Offset.....0x14

Access Mode.....Read/Write

Width .....32 bits

The host uses this register to define the memory base address for the adapter registers.

BIT	BIT NAME	BIT DESCRIPTION
0	MemBaseAddrInd	A value of 1 indicates this register is the memory base address.
2..1	MemMapType	These are read-only bits, and [2] is hard wired to 0. Bit[1] is loaded from EEPROM Lower1Meg bit of the ConfigParm. When set to 01, instructs the host system to map the adapter registers into the lowest 1 megabyte of memory address space. When set to 00, the registers can be map to anywhere within the 32-bit address space.
6..3	Reserved	Reserved for future use. Should be set to 0.
31..7	MemBaseAddress	The system programs the memory base address into this field. Since the adapter registers occupy 128 bytes of I/O space, 25 bits are required to completely specify the memory base address.

MINGNT

Class.....PCI Configuration Registers, Configuration

Base Address **.....PCI device configuration header start**

Address Offset.....0x3e

Access Mode.....Read Only

Width .....8 bits

BIT	BIT NAME	BIT DESCRIPTION
7..0	MinGnt	MinGnt specifies, in 250 ns increments, how long a burst period the adapter requires when operating as a bus master. The value for MinGnt is stored in the ConfigParm word in EEPROM. ST201-based PCI systems return the value 10 in this field. This assumes a 33 MHz bus (30 ns clock period), 1 clock for address phase, 10 clock latency to first data phase, then no wait states for the 64 remaining data phases. (1 + 10 + 64)30ns = 2.25us, rounded up to 2.5us; 2.5us/250 ns = 10

REVISIONID

Class.....PCI Configuration Registers, Configuration

Base Address **.....PCI device configuration header start**

Address Offset.....0x08

Access Mode.....Read Only

Width .....8 bits

BIT	BIT NAME	BIT DESCRIPTION
7..0	RevisionId	This register provides a revision code for the ST201. The first ST201 will return 00h. Future revisions of the chip will cause this value to be incremented.

SUBSYSTEMID

Class.....PCI Configuration Registers, Configuration

Base Address **.....PCI device configuration header start**

Address Offset.....0x2e

Access Mode.....Read Only

Width .....16 bits

BIT	BIT NAME	BIT DESCRIPTION
15..0	SubsystemId	This is the value read from EEPROM word 03h after system reset.

SUBSYSTEMVENDORID

Class.....PCI Configuration Registers, Configuration

Base Address **.....PCI device configuration header start**

Address Offset.....0x2c

Access Mode.....Read Only

Width .....16 bits

BIT	BIT NAME	BIT DESCRIPTION
15..0	SubsystemVen-dorId	This value is read from EEPROM location 02h after system reset.

VENDORID

Class.....PCI Configuration Registers, Configuration

Base Address **.....PCI device configuration header start**

Address Offset.....0x00

Access Mode.....Read Only

Width .....16 bits

BIT	BIT NAME	BIT DESCRIPTION
15..0	VendorId	This register contains the unique 16-bit manufacturer's ID as allocated by the PCI SIG. Sundance's manufacturer ID is hard-wired to 0x13f0.

CAPID

Class.....PCI Configuration Registers, Power Management

Base Address **.....PCI device configuration header start**

Address Offset.....0x50

Access Mode.....Read Only

Width .....8 bits

BIT	BIT NAME	BIT DESCRIPTION
7..0	CapId	This register indicates the type of capability data structure. It returns 01h to indicate a PCI Power Management structure.

NEXTITEMPTR

Class.....PCI Configuration Registers, Power Management

Base Address **.....PCI device configuration header start**

Address Offset.....0x51

Access Mode.....Read Only

Width .....8 bits

BIT	BIT NAME	BIT DESCRIPTION
7..0	NextItemPtr	This register points to the next capability data structure in the capabilities list. It returns 00h to indicate that there are no further data structures.

POWERMGMTCAP

Class.....PCI Configuration Registers, Power Management

Base Address **.....PCI device configuration header start**

Address Offset.....0x52

Access Mode.....Read Only

Width .....16 bits

This register provides information about the adapter's power management capabilities. The reset default is 7601h, but several bits are loaded from EEPROM shortly after reset.

BIT	BIT NAME	BIT DESCRIPTION
2..0	Version	This read-only field returns 1h, as specified in the PCI Bus Power Management Specification Revision 1.0.
8..3	Reserved	Reserved for future use. Should be set to 0.
9	D1Support	This read-only bit, when set, indicates that this device supports the D1 power state. This value of this bit is determined by bit 4 in the EEPROM ConfigParm.
10	D2Support	This read-only bit, when set, indicates that this device supports the D2 power state. This value of this bit is determined by bit 5 in the EEPROM ConfigParm.
15..11	PmeSupport	This read-only field indicates the power states from which this device is able to generate a power management event by asserting PMEN. Each bit corresponds to a power state. A zero in a particular bit indicates that events cannot be generated from that state. The bits are defined as follows: xxxx1: Power management events can be generated from D0. xxx1x: Power management events can be generated from D1. xx1xx: Power management events can be generated from D2. x1xxx: Power management events can be generated from D3hot. 1xxxx: Power management events can be generated from D3Cold. The ST201 hard-wires bit 11 to zero and bit 14 to one. The values of bits 12,13, and 15 are determined by bits 4, 5 and 3 respectively from the EEPROM ConfigParm.

POWERMGMTCTRL

Class.....PCI Configuration Registers, Power Management

Base Address **.....PCI device configuration header start**

Address Offset.....0x54

Access Mode.....Read/Write

Width .....16 bits

This register allows control over the power state and the power management interrupts.

BIT	BIT NAME	BIT DESCRIPTION
1..0	PowerState	This read/write field is used to determine or set the ST201's power state. The following values are defined: 00: State D0 01: State D1 10: State D2 11: State D3 If PowerState is set to a non-zero value, the ST201 will not respond to PCI I/O or memory cycles, nor will it be able to generate PCI bus master cycles.
7..2	Reserved	Reserved for future use. Should be set to 0.
8	PmeEn	When this read/write bit is set, the ST201 is allowed to report wake events on the PMEN signal. The specific events which can generate wake are defined by the WakeEvent I/O register. This bit is loaded from bit[6] of ConfigParm.
14..9	Reserved	Reserved for future use. Should be set to 0.
15	PmeStatus	This read/clear bit is set to indicate a wake event has occurred. This bit is set regardless of the value in PmeEn. Writing a one to this bit clears it. Writing zero has no effect.

EEPROM DATA FORMAT

Figure 13 summarizes the layout of the EEPROM.

byte 1	byte 0	Offset
StationAddress2		0x14
StationAddress1		0x12
StationAddress0		0x10
SubSystemId		0x06
SubSystemVendorId		0x04
AsicCtrl		0x02
ConfigParam		0x00

FIGURE 13: EEPROM Data Layout

CONFIGPARM

Class.....EEPROM Data Format

Base Address **.....0x00, address written to** EepromCtrl **register**

Address Offset.....0x00

Access Mode.....Read Only

Width .....16 bits

This is loaded into the ST201 and controls various hardware functions related to PCI bus operation.

BIT	BIT NAME	BIT DESCRIPTION
0	FastBackToBack	Determines the value for the FastBackToBack bit in the ConfigStatus register.
1	Lower1Meg	Provides the value for MemMapType[0] in the MemBaseAddress register.
2	DisableMemBase	When set, disables the MemBaseAddr register. This bit causes MemBaseAddr to read back as zeroes, appearing as if it is not implemented by the adapter.
3	D3ColdPme	Provides the value returned in bit 15 of the PowerMgmtCap register. This bit, when set, indicates that the adapter is capable of signaling wake from the D3cold state.
4	D1Support	Provides the value returned in D1Support and also bit 12 in the PowerMgmtCap register.
5	D2Support	Provides the value returned in D2Support and also bit 13 in the PowerMgmtCap register.
6	PmeEn	When this bit is set, the ST201 is allowed to report wake events on the PMEN signal.
10..7	MinGnt	Determines the value returned in bits [4..1] of the MinGnt register.
15..11	MaxLat	Determines the value returned in bits [5..1] of the MaxLat register.

ConfigParm has a default value of 2ab0h, but is normally overwritten by the value in EEPROM. The default values specify:

FastBackToBack = 0

Lower1Meg = 0

DisableMemBase = 0

D3ColdPme = 0

D1Support = 1

D2Support = 1

PmeEn = 0

MinGnt = 0101b (2.5 us)

MaxLat = 00101b (2.5 us).

STATIONADDRESS

Class.....EEPROM Data Format

Base Address **.....0x00, address written to** EepromCtrl **register**

Address Offset.....0x10, 0x12, 0x14

Access Mode.....Read Only

Width .....48 bits

This is the field to be programmed into the StationAddress register. OEM customers may choose to program this field with a different value.

BIT	BIT NAME	BIT DESCRIPTION
15..0	StationAddress0	The least significant word of the station address, corresponding to address 0x10.
31..16	StationAddress1	The second least significant word of the station address, corresponding to address 0x12.
47..32	StationAddress2	The most significant word of the station address, corresponding to address 0x14.

ASICCTRL

Class.....EEPROM Data Format

Base Address **.....0x00, address written to** EepromCtrl **register**

Address Offset.....0x02

Access Mode.....Read Only

Width .....16 bits

This word supplies the value for the least significant byte of the AsicCtrl I/O Register. Bit[15] is loaded into WakePolarity of the WakeEvent I/O Register. They are read automatically by the hardware upon reset to provide default settings for non-system related configuration settings. The AsicCtrl register may be over-written by the host system.

BIT	BIT NAME	BIT DESCRIPTION
0	Reserved	Reserved for future use. Should be set to 0.
1	ExpRomSize	Specifies the size of the Expansion ROM installed on the adapter, as follows: 0 = 32 kB (default after reset) 1 = 64 kB
2	TxLargeEnable	This read/write bit, when set, enables transmission of frames that are larger than the TxFIFO. Since ST201's TxFIFO size is 2KB, this bit can be left clear (the reset default).
3	RxLargeEnable	This read/write bit, when set, enables reception of frames that are larger than the RxFIFO. Since ST201's RxFIFO size is 2KB, this bit can be left clear (the reset default).
4	ExpRomDisable	This bit, when set, disables accesses to the on-adapter Expansion ROM. This bit is included to allow bypassing the Expansion ROM without having to physically remove it from the board. When this bit is set, the ST201 responds to any read in its configured Expansion ROM space by returning 00000000h, and it ignores writes to the Expansion ROM. This bit resets to 0.
5	PhySpeed10	This read-only bit, when set, indicates the 10Mb/s operation is available from the PHY on the adapter. "0" indicates the PHY is not 10Mb/s capable.
6	PhySpeed100	This read-only bit, when set, indicates the 100Mb/s operation is available from the PHY on the adapter. "0" indicates the PHY is not 100Mb/s capable.
7	PhyMedia	This read-only bit indicates the media type that is available on the adapter. "0" indicates twisted-pair media, and "1" indicates fiber media. The combination of PhyMedia, PhySpeed100, and PhySpeed10 will determine the capability of the adapter. For example, [7,6,5] = 000: undefined 001: 10BASE-T PHY 010: 100BASE-T PHY 011: 10BASE-T and 100BASE-T dual-speed PHY 100: undefined 101: 10BASE-F PHY 110: 100BASE-F PHY 111: 10BASE-F and 100BASE-F dual-speed PHY

BIT	BIT NAME	BIT DESCRIPTION
14..8	Reserved	Reserved for future use. Should be set to 0.
15	ResetPolarity	Setting this read/write bit will cause the RSTOUT pin to be asserted in the HIGH state (default after RESET).

SUBSYSTEMVENDORID

Address Offset 0x04

Class.....EEPROM Data Format

Access Mode..... Read Only

Base Address **.....0x00, address written to** EepromCtrl **register**

Width..... 16 bits

BIT	BIT NAME	BIT DESCRIPTION
15..0	SubsystemVendorId	This is the two-byte subsystem vendor ID. Since in this case the subsystem is an adapter, customers needs to use their PCI vendor ID.

SUBSYSTEMID

Address Offset 0x06

Class.....EEPROM Data Format

Access Mode..... Read Only

Base Address **.....0x00, address written to** EepromCtrl **register**

Width..... 16 bits

BIT	BIT NAME	BIT DESCRIPTION
15..0	SubsystemId	This is the two-byte subsystem ID for the adapters, the same code as the DeviceId is used.

ABSOLUTE MAXIMUM RATINGS

Storage Temperature .....-65ºC to +150ºC

Ambient Temperature.....-65ºC to +70ºC

Supply Voltage .....-0.3V to +6.0V

Environmental stresses above those listed in Absolute Maximum Ratings may cause permanent damage resulting in device failure. Functionality at or above the limits listed below is not guaranteed. Exposure to the environmental stress at the levels listed below for extended periods may adversely affect device reliability.

OPERATING RANGES

Commercial Devices

Temperature (T_A) ..... 0ºC to +70ºC

Supply Voltages (V_{CC}) .....+5V ±5%

Input voltages.....+5V ±5%

Operating ranges define the limits of guaranteed device functionality.

DC CHARACTERISTICS

DC characteristics are defined over commercial operating ranges unless specified otherwise.

PARAMETER SYMBOL	PARAMETER DESCRIPTION	TEST CONDITIONS	MIN	MAX	UNIT
PIN TYPE IT (TTL, PCI INPUT BUFFER)					
V_{IH}	Input high voltage			2	V
V_{IL}	Input low voltage			0.8	V
I_{IN}	Input leakage current	$V_{IN} = V_{DD}/V_{SS}$	-10	10	µA
PIN TYPE ITU (TTL, PCI INPUT BUFFER WITH PULL UP)					
V_{IH}	Input high voltage			2	V
V_{IL}	Input low voltage			0.8	V
I_{IN}	Input leakage current	$V_{IN} = V_{DD}/V_{SS}$	-10	10	µA
PIN TYPE ITD (TTL, PCI INPUT BUFFER WITH PULL DOWN)					
V_{IH}	Input high voltage			2	V
V_{IL}	Input low voltage			0.8	V
I_{IN}	Input leakage current	$V_{IN} = V_{DD}/V_{SS}$	-10	10	µA
PIN TYPE OT4/OC4 (TTL, CMOS OUTPUT BUFFER)					
V_{OH}	Output high voltage	I_{OH} = -4mA	2.4		V
V_{OL}	Output low voltage	I_{OL} = 4mA		0.4	V
I_{OZ}	Output leakage current	$V_{IN} = V_{DD}/V_{SS}$	-10	10	µA
PIN TYPE OP3 (PCI OUTPUT BUFFER)					
V_{OH}	Output high voltage	I_{OH} = -2mA	2.4		V
V_{OL}	Output low voltage	I_{OL} = 3mA		0.55	V
I_{OZ}	Output leakage current	$V_{IN} = V_{DD}/V_{SS}$	-10	10	µA
PIN TYPE OP6 (PCI OUTPUT BUFFER FOR PCI I/O WITH PULL UP)					
V_{OH}	Output high voltage	I_{OH} = -2mA	2.4		V
V_{OL}	Output low voltage	I_{OL} = 6mA		0.55	V
I_{OZ}	Output leakage current	$V_{IN} = V_{DD}/V_{SS}$	-10	10	µA

TABLE 4: DC Characteristics

PARAMETER SYMBOL	PARAMETER DESCRIPTION	TEST CONDITIONS	MIN	MAX	UNIT
PIN TYPE OD6 (OPEN DRAIN OUTPUT BUFFER)					
V_{OL}	Output low voltage	I_{OL} = 6mA		0.4	V
I_{OZ}	Output leakage current	$V_{IN} = V_{DD}/V_{SS}$	-10	10	µA
PIN TYPE OD8 (OPEN DRAIN OUTPUT BUFFER)					
V_{OL}	Output low voltage	I_{OL} = 8mA		0.4	V
I_{OZ}	Output leakage current	$V_{IN} = V_{DD}/V_{SS}$	-10	10	µA

TABLE 4: DC Characteristics

PIN TYPE	PINS
PCI INTERFACE	
IT	RSTN, PCICLK, GNTN, IDSEL
IT/OP3	AD[31:0], CBEN[3:0], PAR
ITU/OP6	FRAMEN, IRDYN, TRDYN, DEVSELN, STOPN, PERRN
OD6	INTAN, PMEN, SERRN
OP3	REQN
EXPANSIONROM INTERFACE	
ITU/OT4	ED[7:5], ED[3:0]
ITD/OT4	ED[4]
OT4	EWEN, EOEN, EA[15:0]
EEPROM INTERFACE	
IT	EEDO
OT4	EEDI, EESK, EECS
MII INTERFACE	
IT	TXCLK, CRS, COL, RXER, RXDV, RXD[3:0], RXCLK
IT	PHYLNKN, PHYSPDN, PHYDPLXN
OT4	TXD[3:0], TXEN, MDC
IT/OT4	MDIO

TABLE 5: Pin Type Assignment

PIN TYPE	PINS
MISC INTERFACE	
ITU/OT4	GPIO0, GPIO1
OT4	RSTOUT
OD8	LEDPWRN, LEDLNKN, LEDDPLXN, LEDSPDN
OC4	CLK25
OSCI	X25I
OSCOH1	X25O

TABLE 5: Pin Type Assignment

SWITCHING CHARACTERISTICS

PARAMETER SYMBOL	PARAMETER DESCRIPTION	TEST CONDITIONS	MIN	MAX	UNIT
PCI INTERFACE					
T_{rc}	RSTN cycle		300	-	-
T_{cc}	PCICLK cycle		30	-	-
T_{ch}	PCICLK high		11	-	-
T_{cl}	PCICLK low		11	-	-
T_{rv}	PCICLK rise to bused signal valid		2	11	-
T_{rvp}	PCICLK rise to REQN, GNTN valid		2	12	-
T_{rzo}	PCICLK rise to signal on		2	-	-
T_{roz}	PCICLK rise to signal off		-	28	-
T_{su}	bused signal setup wrt PCI-CLK rise		7	-	-
T_{sup1}	GNTN setup wrt PCICLK rise		10	-	-
T_{sup2}	REQN setup wrt PCICLK rise		12	-	-
T_{hd}	signal hold wrt PCICLK rise		0	-	-
T_{rstoff}	RSTN low to output signal float		-	40	-
EXPANSION ROM INTERFACE - READ					
T_{adv}	ED valid from EA stable		0	150	-
T_{odv}	ED valid from EOEN low		0	70	-
T_{dvz}	ED tri-stated from EOEN high		0	40	-
EXPANSION ROM INTERFACE - LOAD					
T_{as}/T_{os}	EA, EOEN setup wrt EWEN fall		0		-
T_{ah}	EA hold wrt EWEN fall		50	-	-
T_{ds}	ED setup wrt EWEN rise		35	-	-
T_{dh}/T_{oh}	ED, EOEN hold wrt EWEN fall		0		-

TABLE 6: Switching Characteristics

PARAMETER SYMBOL	PARAMETER DESCRIPTION	TEST CONDITIONS	MIN	MAX	UNIT
T_{wh}	EWEN write cycle high		100	-	-
T_{wl}	EWEN write cycle low		90	-	-
EEPROM INTERFACE					
T_{skc}	EESK cycle		1us	-	-
T_{skh}	EESK high		250	-	-
T_{skl}	EESK low		250	-	-
T_{cs}	EECS low		250	-	-
T_{pd}	EEDI valid wrt EESK rise		100	-	-
T_{csk}	EECS setup wrt EESK rise		50	-	-
T_{csh}	EECS hold wrt EESK fall		0	-	-
T_{dos}	EEDO setup wrt EESK rise		70	500	-
T_{doh}	EEDO hold wrt EESK rise		-	500	-
MII INTERFACE - TRANSMIT					
T_{cc}	TXCLK cycle		-	-	40T
T_{ch}	TXCLK high		14T	26T	-
T_{cl}	TXCLK low	T = 1 when 100Mb/s; 10 when 10Mb/s	14T	26T	-
T_{rv}	TXCLK rise to TXD, TXEN valid			20	
T_{rh}	TXD, TXEN hold after TXCLK rise		5	-	-
MII INTERFACE - RECEIVE					
T_{cc}	RXCLK cycle		-	-	40T
T_{ch}	RXCLK high		14T	26T	-
T_{cl}	RXCLK low	T = 1 when 100Mb/s; 10 when 10Mb/s	14T	26T	-
T_{su}	RXD,RXER,RXDV setup wrt RXCLK rise		10	-	-
T_{hd}	RXD,RXER,RXDV hold wrt RXCLK rise		5	-	-

TABLE 6: Switching Characteristics

PARAMETER SYMBOL	PARAMETER DESCRIPTION	TEST CONDITIONS	MIN	MAX	UNIT
MII INTERFACE - MANAGEMENT					
T_{cc}	MDC cycle		400	-	-
T_{ch}	MDC high		160	-	-
T_{cl}	MDC low		160	-	-
T_{su}	MDIO setup wrt MDC rise		10	-	-
T_{hd}	MDIO hold wrt MDC rise		10	-	-
T_{rv}	MDC rise to MDIO valid		-	20	-
MISC INTERFACE					
T_{cc}	CLK25 cycle		-	-	40
T_{ch}	CLK25 high		16	24	-
T_{cl}	CLK25 low		16	24	-

TABLE 6: Switching Characteristics

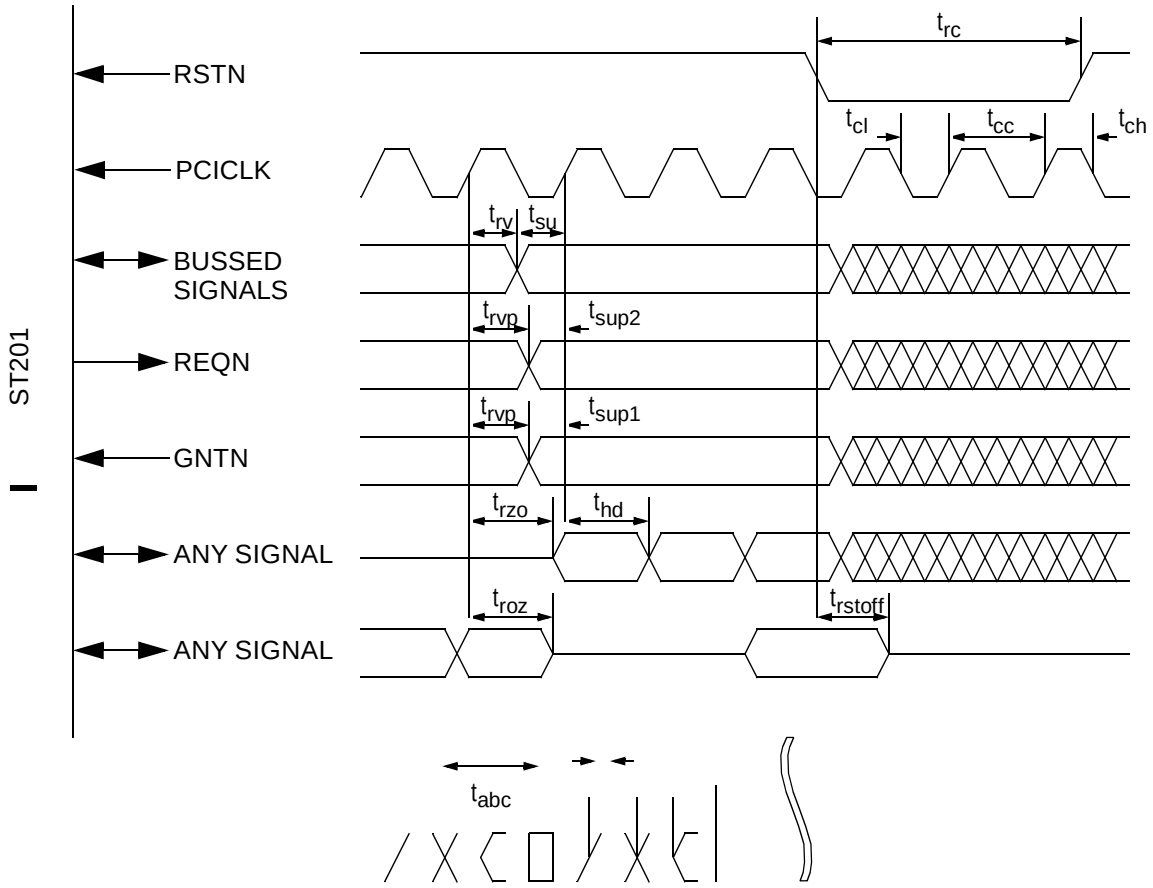


FIGURE 14: PCI Switching Characteristics

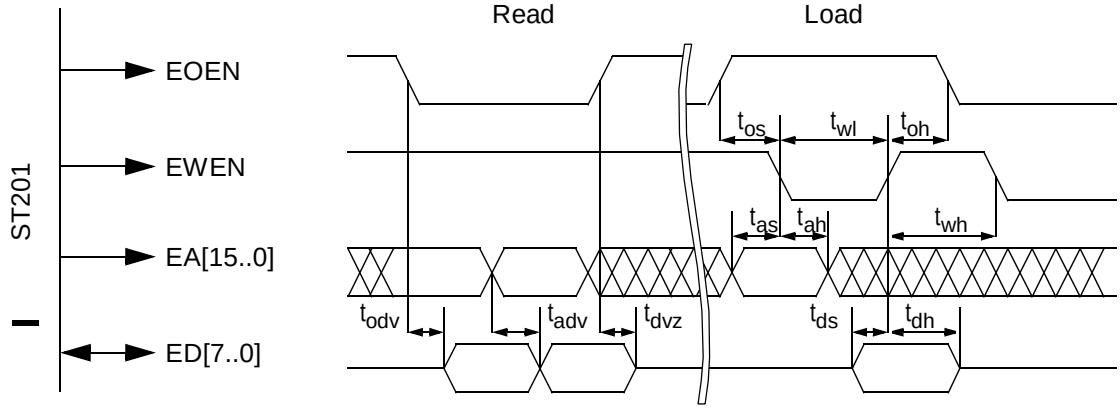


FIGURE 15: Expansion ROM Switching Characteristics

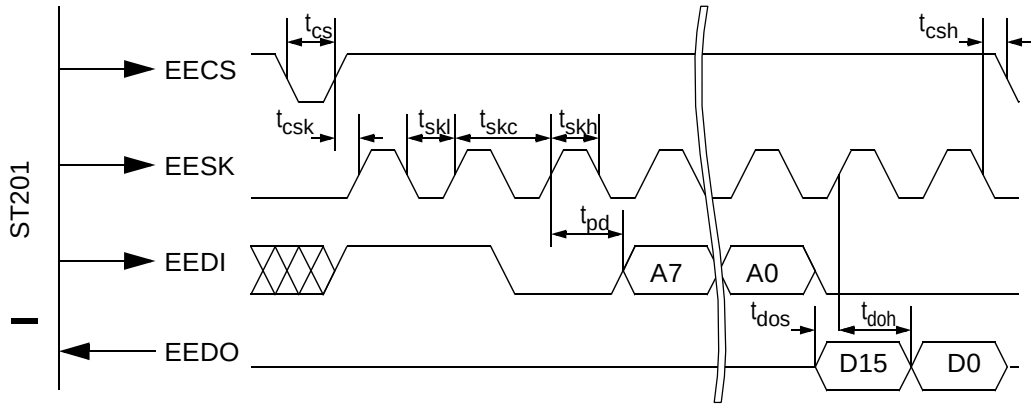


FIGURE 16: EEPROM Switching Characteristics

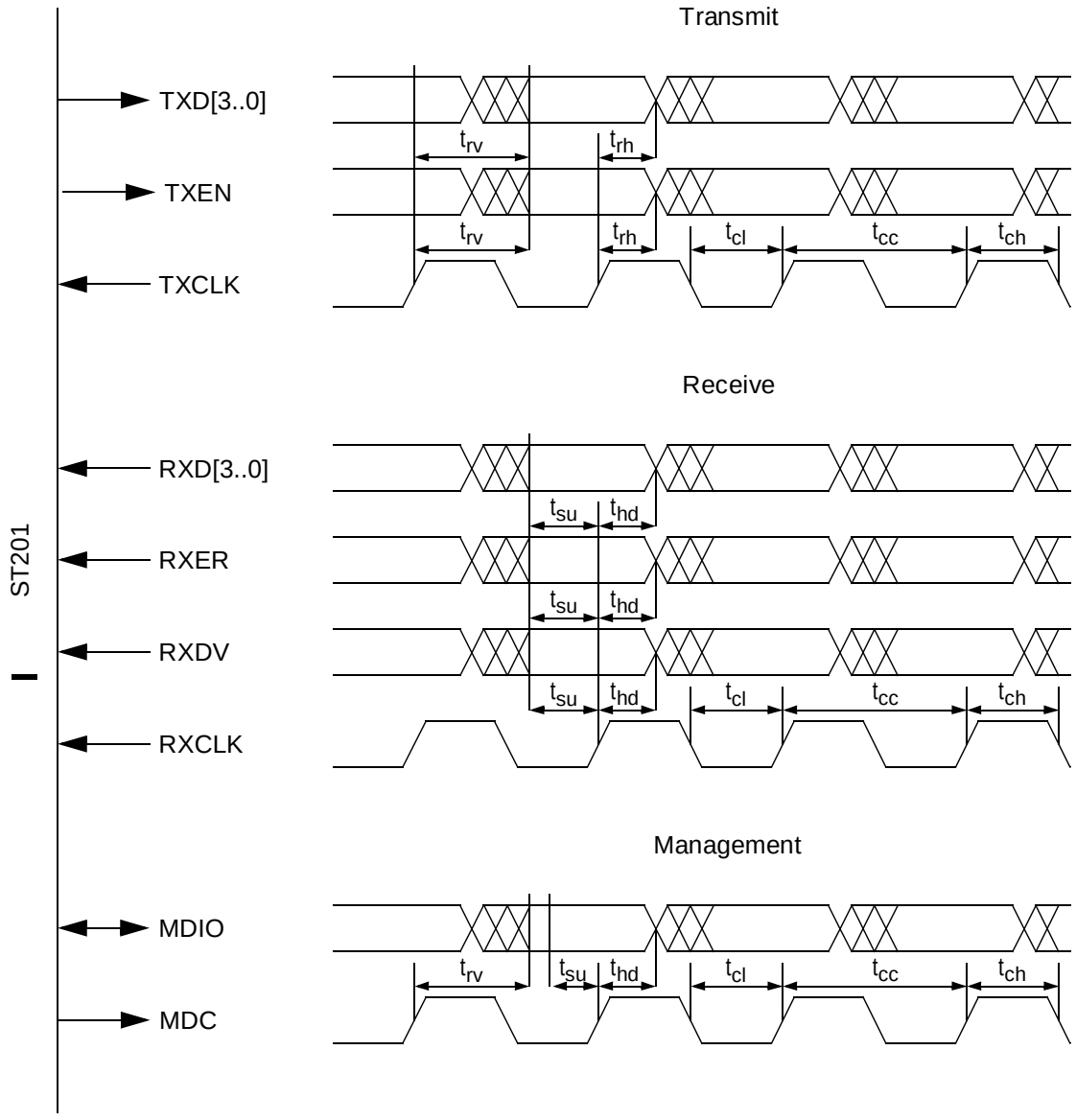


FIGURE 17: MII Switching Characteristics

PHYSICAL DIMENSIONS