Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!

Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832
Email & Skype: info@chipsmall.com Web: www.chipsmall.com
Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China

LAN9220

16-bit Non-PCI Small Form Factor 10/100 Ethernet Controller with Variable Voltage I/O & HP Auto-MDIX Support

Highlights

- Efficient architecture with low CPU overhead
- Easily interfaces to most 16-bit embedded CPU's
- 1.8V to 3.3V variable voltage I/O accommodates wide range of I/O signalling without voltage level shifters
- Integrated PHY with HP Auto-MDIX support
- Integrated checksum offload engine helps reduce CPU load
- Low pin count and small body size package for small form factor system designs

Target Applications

- Cable, satellite, and IP set-top boxes
- Digital video recorders
- Video-over IP solutions, IP PBX & video phones
- Wireless routers & access points
- Audio distribution systems
- Printers, kiosks, security systems
- General embedded applications

Key Benefits

- Non-PCI Ethernet controller for standard performance applications
  - 16-bit interface
  - Burst-mode read support
- Minimizes dropped packets
  - Internal buffer memory can store over 200 packets
  - Automatic PAUSE and back-pressure flow control
- Minimizes CPU overhead
  - Supports Slave-DMA
  - Interrupt Pin with Programmable Hold-off timer
- Reduces system cost and increases design flexibility
- SRAM-like interface easily interfaces to most embedded CPU's or SoC's
- Reduced Power Modes
  - Numerous power management modes
  - Wake on LAN
  - Magic packet wakeup
  - Wakeup indicator event signal
  - Link Status Change
- Single chip Ethernet controller
  - Fully compliant with IEEE 802.3/802.3u standards
  - Integrated Ethernet MAC and PHY
  - 10BASE-T and 100BASE-TX support
  - Full- and Half-duplex support
  - Full-duplex flow control
  - Backpressure for half-duplex flow control
  - Preamble generation and removal
  - Automatic 32-bit CRC generation and checking
  - Automatic payload padding and pad removal
  - Loop-back modes
- Flexible address filtering modes
  - One 48-bit perfect address
  - 64 hash-filtered multicast addresses
  - Pass all multicast
  - Promiscuous mode
  - Inverse filtering
  - Pass all incoming with status report
  - Disable reception of broadcast packets
- Integrated 10/100 Ethernet PHY
  - Supports HP Auto-MDIX
  - Auto-negotiation
  - Supports energy-detect power down
- Host bus interface
  - Simple, SRAM-like interface
  - 16-bit data bus
  - 16Kbyte FIFO with flexible TX/RX allocation
  - One configurable host interrupt
- Miscellaneous features
  - Small form factor, 56-pin QFN RoHS Compliant package
  - Integrated 1.8V regulator
  - Integrated checksum offload engine
  - Mixed endian support
  - General Purpose Timer
  - Optional EEPROM interface
  - Support for 3 status LEDs multiplexed with Programmable GPIO signals
- Single 3.3V Power Supply with Variable Voltage I/O
- 0°C to +70°C Commercial Temperature Support

TO OUR VALUED CUSTOMERS

It is our intention to provide our valued customers with the best documentation possible to ensure successful use of your Microchip products. To this end, we will continue to improve our publications to better suit your needs. Our publications will be refined and enhanced as new volumes and updates are introduced.

If you have any questions or comments regarding this publication, please contact the Marketing Communications Department via E-mail at **docerrors@microchip.com**. We welcome your feedback.

Most Current Data Sheet

To obtain the most up-to-date version of this data sheet, please register at our Worldwide Web site at:

http://www.microchip.com

You can determine the version of a data sheet by examining its literature number found on the bottom outside corner of any page. The last character of the literature number is the version number, (e.g., DS30000000A is version A of document DS30000000).

Errata

An errata sheet, describing minor operational differences from the data sheet and recommended workarounds, may exist for current devices. As device/documentation issues become known to us, we will publish an errata sheet. The errata will specify the revision of silicon and revision of document to which it applies.

To determine if an errata sheet exists for a particular device, please check with one of the following:

- Microchip's Worldwide Web site; http://www.microchip.com
- Your local Microchip sales office (see last page)

When contacting a sales office, please specify which device, revision of silicon and data sheet (include -literature number) you are using.

Customer Notification System

Register on our web site at **www.microchip.com** to receive the most current information on all of our products.

Table of Contents

1.0 GENERAL DESCRIPTION

The LAN9220 is a full-featured, single-chip 10/100 Ethernet controller designed for embedded applications where performance, flexibility, ease of integration and system cost control are required. The LAN9220 is fully IEEE 802.3 10BASE-T and 802.3u 100BASE-TX compliant, and supports HP Auto-MDIX. The variable voltage I/O signals of the LAN9220 accommodate lower voltage I/O signalling without the need for voltage level shifters.

The LAN9220 includes an integrated Ethernet MAC and PHY with a high-performance SRAM-like slave interface. The simple, yet highly functional host bus interface provides a glue-less connection to most common 16-bit microprocessors and microcontrollers as well as 32-bit microprocessors with a 16-bit external bus. The integrated checksum offload engines enable the automatic generation of the 16-bit checksum for received and transmitted Ethernet frames, offloading the task from the CPU. The LAN9220 also includes large transmit and receive data FIFOs to accommodate high latency applications. In addition, the LAN9220 memory buffer architecture allows highly efficient use of memory resources by optimizing packet granularity.

Applications

The LAN9220 is well suited for many embedded applications, including:

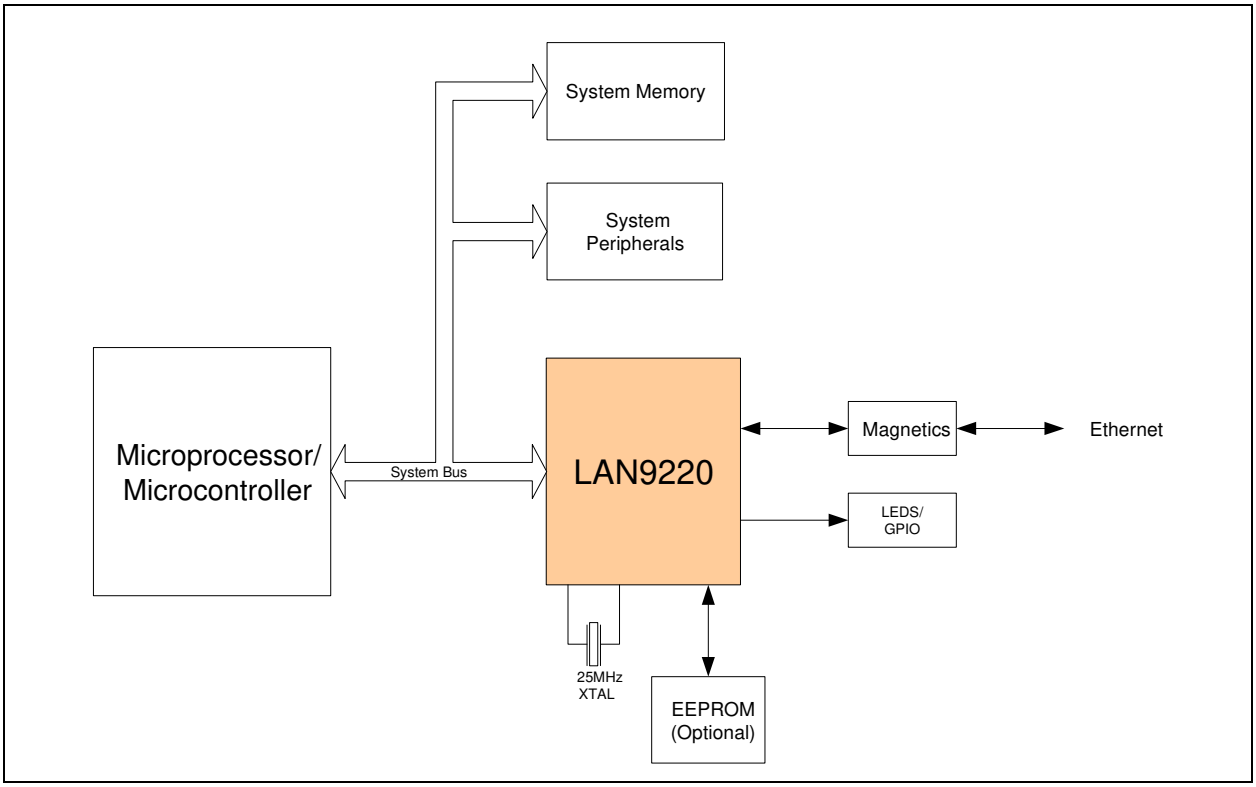
- Printers, kiosks, POS terminals and security systems
- Audio distribution systems
- General embedded systems
- Basic cable, satellite and IP set-top boxes
- Voice-over-IP solutions

The LAN9220 also supports features which reduce or eliminate packet loss. Its internal 16-KByte SRAM can hold over 200 received packets. If the receive FIFO gets too full, the LAN9220 can automatically generate flow control packets to the remote node, or assert back-pressure on the remote node by generating network collisions.

The LAN9220 supports numerous power management and wakeup features. The LAN9220 can be placed in a reduced power mode and can be programmed to issue an external wake signal via several methods, including “Magic Packet”, “Wake on LAN” and “Link Status Change”. This signal is ideal for triggering system power-up using remote Ethernet wakeup events. The device can be removed from the low power state via a host processor command.

1.1 Block Diagram

FIGURE 1-1: SYSTEM BLOCK DIAGRAM



The Microchip LAN9220 integrated 10/100 MAC/PHY controller is a peripheral chip that performs the function of translating parallel data from a host controller into Ethernet packets. The LAN9220 Ethernet MAC/PHY controller is designed and optimized to function in an embedded environment. All communication is performed with programmed I/O transactions using the simple SRAM-like host interface bus.

The diagram shown above, describes a typical system configuration of the LAN9220 in a typical embedded environment.

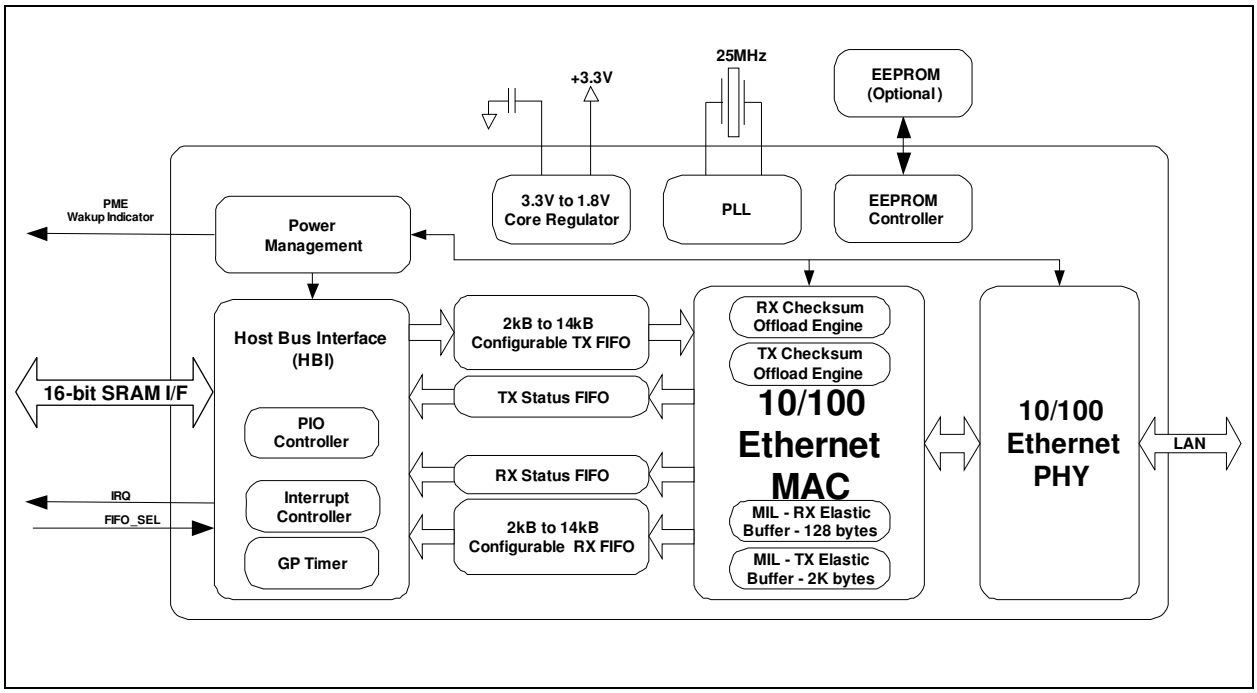
The LAN9220 is a general purpose, platform independent, Ethernet controller. The LAN9220 consists of four major functional blocks. The four blocks are:

- 10/100 Ethernet PHY
- 10/100 Ethernet MAC
- RX/TX FIFOs
- Host Bus Interface (HBI)

1.2 Internal Block Overview

This section provides an overview of each of these functional blocks as shown in Figure 1-2, "Internal Block Diagram".

FIGURE 1-2: INTERNAL BLOCK DIAGRAM



1.3 10/100 Ethernet PHY

The LAN9220 integrates an IEEE 802.3 physical layer for twisted pair Ethernet applications. The PHY can be configured for either 100 Mbps (100BASE-TX) or 10 Mbps (10BASE-T) Ethernet operation in either full or half duplex configurations. The PHY block supports HP Auto-MDIX and auto-negotiation.

Minimal external components are required for the utilization of the Integrated PHY.

1.4 10/100 Ethernet MAC

The transmit and receive data paths are separate within the MAC allowing the highest performance especially in full duplex mode. The data paths connect to the PIO interface Function via separate busses to increase performance. Payload data as well as transmit and receive status is passed on these busses.

A third internal bus is used to access the MAC's Control and Status Registers (CSR's). This bus is accessible from the host through the PIO interface function.

On the backend, the MAC interfaces with the internal 10/100 PHY through a MII (Media Independent Interface) port internal to the LAN9220. The MAC CSR's also provide a mechanism for accessing the PHY's internal registers through the internal SMI (Serial Management Interface) bus.

The MAC Interface Layer (MIL), within the MAC, contains a 2K Byte transmit and a 128 Byte receive FIFO which is separate from the TX and RX FIFOs. The FIFOs within the MAC are not directly accessible from the host interface. The differentiation between the TX/RX FIFO memory buffers and the MAC buffers is that when the transmit or receive packets are in the MAC buffers, the host no longer can control or access the TX or RX data. The MAC buffers (both TX and RX) are in effect the working buffers of the Ethernet MAC logic. In the case of reception, the data must be moved first to the RX FIFOs for the host to access the data. For TX operations, the MIL operates in store-and-forward mode and will queue an entire frame before beginning transmission.

1.5 Receive and Transmit FIFOs

The Receive and Transmit FIFOs allow increased packet buffer storage to the MAC. The FIFOs are a conduit between the host interface and the MAC through which all transmitted and received data and status information is passed. Deep FIFOs allow a high degree of latency tolerance relative to the various transport and OS software stacks thus reducing or minimizing overrun conditions. Like the MAC, the FIFOs have separate receive and transmit data paths. In addition, the RX and TX FIFOs are configurable in size, allowing increased flexibility.

1.6 Interrupt Controller

The LAN9220 supports a single programmable interrupt. The programmable nature of this interrupt allows the user the ability to optimize performance dependent upon the application requirement. Both the polarity and buffer type of the interrupt pin are configurable for the external interrupt processing. The interrupt line can be configured as an open-drain output to facilitate the sharing of interrupts with other devices. In addition, a programmable interrupt de-assertion interval is provided.

1.7 GPIO Interface

A 3-bit GPIO and 2-bit GPO (Multiplexed on the EEPROM and LED Pins) interface is included in the LAN9220. It is accessible through the host bus interface via the CSRs. The GPIO signals can function as inputs, push-pull outputs and open drain outputs. The GPIO's (GPO's are not configurable) can also be configured to trigger interrupts with programmable polarity.

1.8 Serial EEPROM Interface

A serial EEPROM interface is included in the LAN9220. The serial EEPROM is optional and can be programmed with the LAN9220 MAC address. The LAN9220 can optionally load the MAC address automatically after hardware reset, or soft reset.

1.9 Power Management Controls

The LAN9220 supports comprehensive array of power management modes to allow use in power sensitive applications. Wake on LAN, Link Status Change and Magic Packet detection are supported by the LAN9220. An external PME (Power Management Event) interrupt is provided to indicate detection of a wakeup event.

1.10 General Purpose Timer

The general-purpose timer has no dedicated function within the LAN9220 and may be programmed to issue a timed interrupt.

1.11 Host Bus Interface (SRAM Interface)

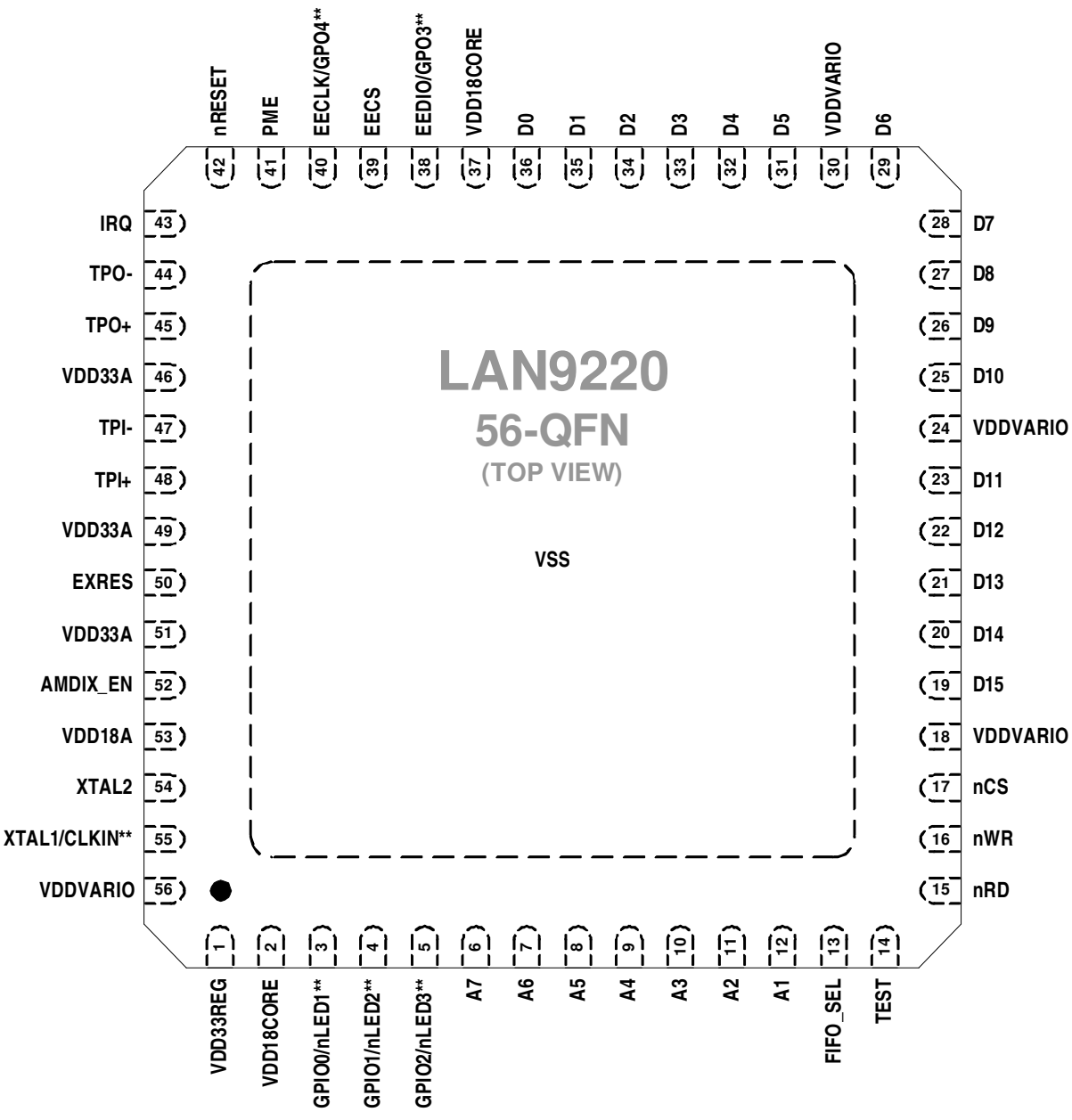
The host bus interface provides a FIFO interface for the transmit and receive data paths, as well as an interface for the LAN9220 Control and Status Registers (CSR's).

The host bus interface is the primary bus for connection to the embedded host system. This interface models an asynchronous SRAM. TX FIFO, RX FIFO, and CSR's are accessed through this interface. Programmed I/O transactions are supported.

The LAN9220 host bus interface supports 16-bit bus transfers. Internally, all data paths are 32-bits wide. The LAN9220 can be interfaced to either Big-Endian or Little-Endian processors and includes mixed endian support for FIFO accesses.

2.0 PIN DESCRIPTION AND CONFIGURATION

FIGURE 2-1: 56-QFN PIN CONFIGURATION (TOP VIEW)



**DENOTES A MULTIFUNCTION PIN

NOTE: When HP Auto-MDIX is activated, the TPO+/- pins can function as TPI +/- and vice-versa

NOTE: Exposed pad (VSS) on bottom of package must be connected to ground

2.1 Pin List

TABLE 2-1: HOST BUS INTERFACE SIGNALS

Name	Symbol	Buffer Type	# Pins	Description
Host Data	D[15:0]	VIS/VO8	16	Bi-directional data port.
Host Address	A[7:1]	VIS	7	7-bit Address Port. Used to select Internal CSR's and TX and RX FIFOs.
Read Strobe	nRD	VIS	1	Active low strobe to indicate a read cycle.
Write Strobe	nWR	VIS	1	Active low strobe to indicate a write cycle. This signal, qualified with nCS, is also used to wakeup the LAN9220 when it is in a reduced power state.
Chip Select	nCS	VIS	1	Active low signal used to qualify read and write operations. This signal qualified with nWR is also used to wakeup the LAN9220 when it is in a reduced power state.
Interrupt Request	IRQ	VO8/ VOD8	1	Programmable Interrupt request. Programmable polarity, source and buffer types.
FIFO Select	FIFO_SEL	VIS	1	When driven high all accesses to the LAN9220 are to the RX or TX Data FIFOs. In this mode, the A[7:3] upper address inputs are ignored.

TABLE 2-2: LAN INTERFACE SIGNALS

Name	Symbol	Buffer Type	# Pins	Description
TPO+	TPO+	AO	1	Transmit Positive Output (normal) Receive Positive Input (reversed)
TPO-	TPO-	AO	1	Transmit Negative Output (normal) Receive Negative Input (reversed)
TPI+	TPI+	AI	1	Receive Positive Input (normal) Transmit Positive Input (reversed)
TPI-	TPI-	AI	1	Receive Negative Input (normal) Transmit Negative Output (reversed)
PHY External Bias Resistor	EXRES	AI	1	Must be connected to ground through a 12.4K ohm 1% resistor.

Note: The pin names for the twisted pair pins shown above apply to a normal connection. If HP Auto-MDIX is enabled and a reverse connection is detected, or a reverse connection is manually selected, the input pins become outputs, and vice-versa, as indicated in the descriptions.

TABLE 2-3: SERIAL EEPROM INTERFACE SIGNALS

Name	Symbol	Buffer Type	# Pins	Description
EEPROM Data, GPO3, TX_EN, TX_CLK	EEDIO/GPO3/ TX_EN/TX_CLK	VIS/VO8	1	**EEPROM Data:** This bi-directional pin can be connected to a serial EEPROM DIO. This is optional. **General Purpose Output 3:** This pin can also function as a general purpose output, or it can be configured to monitor the TX_EN or TX_CLK signals on the internal MII port. When configured as a GPO signal, or as a TX_EN/TX_CLK monitor, the EECS pin is deasserted so as to never unintentionally access the serial EEPROM. This signal cannot function as a general-purpose input.
EEPROM Chip Select	EECS	VO8	1	Serial EEPROM chip select.
EEPROM Clock, GPO4 RX_DV, RX_CLK	EECLK/GPO4/ RX_DV/RX_CLK	VO8 (PU)	1	**EEPROM Clock:** Serial EEPROM Clock pin. **General Purpose Output 4:** This pin can also function as a general-purpose output, or it can be configured to monitor the RX_DV or RX_CLK signals on the internal MII port. When configured as a GPO signal, or as an RX_DV/RX_CLK monitor, the EECS pin is deasserted so as to never unintentionally access the serial EEPROM. This signal cannot function as a general-purpose input. **Note:** When the EEPROM interface is not used, the EECLK pin must be left unconnected. **Note:** When operating at reduced VDDVARIO voltage levels (less than 3.0V), this pin must be pulled-up with an external resistor. Refer to Section 2.2, "External Pull-Up/Pull-Down Resistors" for more information. **Note:** This pin must not be pulled low by an external resistor or driven low externally under any conditions.

TABLE 2-4: SYSTEM AND POWER SIGNALS

Name	Symbol	Buffer Type	# Pins	Description
Crystal 1, Clock In	XTAL1/CLKIN	ICLK	1	External 25MHz Crystal Input. This pin can also be connected to single-ended TTL oscillator (CLKIN). If this method is implemented, XTAL2 should be left unconnected.
Crystal 2	XTAL2	OCLK	1	External 25MHz Crystal output.
Reset	nRESET	VIS (PU)	1	Active-low reset input. Resets all logic and registers within the LAN9220. This signal is pulled high with a weak internal pull-up resistor. **Note:** The LAN9220 must be reset on power-up via nRESET or following power-up via a soft reset (SRST). The LAN9220 must always be read at least once after reset, or upon return from a power-saving state or write operations will not function. See Section 3.11, "Detailed Reset Description," on page 39 for additional information **Note:** When operating at reduced VDDVARIO voltage levels (less than 3.0V), this pin must be pulled-high to a valid level with an external resistor or must be driven as an input. Refer to Section 2.2, "External Pull-Up/Pull-Down Resistors" for more information.
Wakeup Indicator	PME	VO8/ VOD8	1	When programmed to do so, is asserted when the LAN9220 detects a wake event and is requesting the system to wake up from the associated sleep state. The polarity and buffer type of this signal is programmable. **Note:** Detection of a Power Management Event, and assertion of the PME signal will not wakeup the LAN9220. The LAN9220 will only wake up when it detects a host write cycle (assertion of nCS and nWR). Although any write to the LAN9220, regardless of the data written, will wake-up the device when it is in a power-saving mode, it is required that the BYTE_TEST register be used for this purpose.

TABLE 2-4: SYSTEM AND POWER SIGNALS (CONTINUED)

Name	Symbol	Buffer Type	# Pins	Description
Auto-MDIX Enable	AMDIX_EN	VIS (PU)	1	Enables Auto-MDIX. Pull high or leave unconnected to enable Auto-MDIX, pull low to disable Auto-MDIX. **Note:** When operating at reduced VDDVARIO voltage levels (less than 3.0V), this pin must be pulled to a valid level with an external resistor. Refer to Section 2.2, "External Pull-Up/Pull-Down Resistors" for more information.
Test	TEST	VIS (PD)	1	Reserved for internal test purposes only. **Note:** When operating at a reduced VDDVARIO voltage (less than 3.0V), this pin must be connected to ground or pulled-low with an external resistor. When VDDVARIO = 3.3V, this pin may be left unconnected. Refer to Section 2.2, "External Pull-Up/Pull-Down Resistors" for more information.
General Purpose I/O data, nLED1 (Speed Indicator), nLED2 (Link & Activity Indicator), nLED3 (Full-Duplex Indicator).	GPIO[2:0]/ nLED[3:1]	VIS/ VO12/ VOD12	3	**General Purpose I/O data:** These three general-purpose signals are fully programmable as either push-pull output, open-drain output or input by writing the GPIO_CFG configuration register in the CSR's. They are also multiplexed as GP LED connections. GPIO signals are Schmitt-triggered inputs. When configured as LED outputs these signals are open-drain. **nLED1 (Speed Indicator).** This signal is driven low when the operating speed is 100Mb. During auto-negotiation, when the cable is disconnected, and during 10Mbs operation, this signal is driven high. **nLED2 (Link & Activity Indicator).** This signal is driven low (LED on) when the LAN9220 detects a valid link. This signal is pulsed high (LED off) for 80mS whenever transmit or receive activity is detected. This signal is then driven low again for a minimum of 80mS, after which time it will repeat the process if TX or RX activity is detected. Effectively, LED2 is activated solid for a link. When transmit or receive activity is sensed LED2 will flash as an activity indicator. **nLED3 (Full-Duplex Indicator).** This signal is driven low when the link is operating in full-duplex mode.
Variable Voltage I/O Power	VDDVARIO	P	4	Variable Voltage I/O logic power supply pins. Refer to Section 7.2, "Operating Conditions**," on page 126 for additional details.
Common Ground	VSS	P	1 pad	Common Ground

TABLE 2-4: SYSTEM AND POWER SIGNALS (CONTINUED)

Name	Symbol	Buffer Type	# Pins	Description
+3.3V Regulator Power Supply	VDD33REG	P	1	+3.3V power supply for internal +1.8V regulator.
+3.3V Analog Power	VDD33A	P	3	+3.3V analog power supply pins.
+1.8V Analog Power	VDD18A	P	1	+1.8V analog power supply pin. This pin must be connected externally to VDD18CORE.
Core Voltage Decoupling	VDD18CORE	P	2	+1.8 V from internal core regulator. Both pins must be connected together externally. Each pin requires a 0.01uF decoupling capacitor. In addition, pin 2 requires a bulk 4.7uF capacitor (<2 Ohm ESR) in parallel. These pins must not be used to supply power to other external devices.

TABLE 2-5: 56-QFN PACKAGE PIN ASSIGNMENTS

Pin Num	Pin Name	Pin Num	Pin Name	Pin Num	Pin Name	Pin Num	Pin Name
1	VDD33REG	15	nRD	29	D6	43	IRQ
2	VDD18CORE	16	nWR	30	VDDVARIO	44	TPO-
3	GPIO0/nLED1	17	nCS	31	D5	45	TPO+
4	GPIO1/nLED2	18	VDDVARIO	32	D4	46	VDD33A
5	GPIO2/nLED3	19	D15	33	D3	47	TPI-
6	A7	20	D14	34	D2	48	TPI+
7	A6	21	D13	35	D1	49	VDD33A
8	A5	22	D12	36	D0	50	EXRES
9	A4	23	D11	37	VDD18CORE	51	VDD33A
10	A3	24	VDDVARIO	38	EEDIO/GPO3	52	AMDIX_EN
11	A2	25	D10	39	EECS	53	VDD18A
12	A1	26	D9	40	EECLK/GPO4	54	XTAL2
13	FIFO_SEL	27	D8	41	PME	55	XTAL1/CLKIN
14	TEST	28	D7	42	nRESET	56	VDDVARIO
EXPOSED PAD MUST BE CONNECTED TO VSS							

2.2 External Pull-Up/Pull-Down Resistors

As detailed in Table 2-6, when using an external pull-up or pull-down resistor, the required value is dependent on the operating voltage. Usage of recommended pull-up/pull-down resistor values is required to ensure proper operation.

TABLE 2-6: EXTERNAL PULL-UP/PULL-DOWN RESISTOR VALUES

I/O Voltage	Pull-Up/Pull-Down Resistor Value (Ohms)
3.3V +/- 300mV	10K
2.5 +/- 10%	7.5K
1.8V +/- 10%	4.7K

2.3 Buffer Types

TABLE 2-7: BUFFER TYPES

Type	Description
VIS	Variable voltage input pin
VO12	Variable voltage output with 12mA sink and 12mA source (Note 2-1)
VOD12	Variable voltage open-drain output with 12mA sink (Note 2-1)
VO8	Variable voltage output 8mA symmetrical drive (Note 2-1)
VOD8	Variable voltage open-drain output with 8 mA sink (Note 2-1)
PU	50uA (typical) internal pull-up (Note 2-2)
PD	50uA (typical) internal pull-down (Note 2-2)
AI	Analog input
AO	Analog output
AIO	Analog bi-directional
ICLK	Crystal oscillator input pin
OCLK	Crystal oscillator output pin

Note 2-1 Sink and source capabilities are dependent on the VDDVARIO voltage. Refer to Section 7.6, "DC Electrical Specifications" for additional information.

Note 2-2 When operating at reduced VDDVARIO voltage levels (less than 3.0V), do not rely on internal pull-up and pull-down resistors to determine signal state. Refer to Section 2.2, "External Pull-Up/Pull-Down Resistors" and the individual signal descriptions in Section 2.1, "Pin List" for more information.

3.0 FUNCTIONAL DESCRIPTION

3.1 10/100 Ethernet MAC

The Ethernet Media Access controller (MAC) incorporates the essential protocol requirements for operating an Ethernet/IEEE 802.3-compliant node and provides an interface between the host subsystem and the internal Ethernet PHY. The MAC can operate in either 100-Mbps or 10-Mbps mode.

The MAC operates in both half-duplex and full-duplex modes. When operating in half-duplex mode, the MAC complies fully with Section 4 of ISO/IEC 8802-3 (ANSI/IEEE standard) and ANSI/IEEE 802.3 standards. When operating in full-duplex mode, the MAC complies with IEEE 802.3x full-duplex operation standard.

The MAC provides programmable enhanced features designed to minimize host supervision, bus utilization, and pre- or post-message processing. These features include the ability to disable retries after a collision, dynamic FCS (Frame Check Sequence) generation on a frame-by-frame basis, automatic pad field insertion and deletion to enforce minimum frame size attributes, layer 3 checksum calculation for transmit and receive operations, and automatic retransmission and detection of collision frames.

The MAC can sustain transmission or reception of minimally-sized back-to-back packets at full line speed with an inter-packet gap (IPG) of 9.6 microseconds for 10 Mbps and 0.96 microseconds for 100 Mbps.

The primary attributes of the MAC Function are:

- Transmit and receive message data encapsulation
- Framing (frame boundary delimitation, frame synchronization)
- Error detection (physical medium transmission errors)
- Media access management
- Medium allocation (collision detection, except in full-duplex operation)
- Contention resolution (collision handling, except in full-duplex operation)
- Flow control during full-duplex mode
- Decoding of control frames (PAUSE command) and disabling the transmitter
- Generation of control frames
- Interface to the internal PHY.
- Checksum offload engine for calculation of layer 3 transmit and receive checksum.

The transmit and receive data paths are separate within the LAN9220 from the MAC to host interface allowing the highest performance, especially in full duplex mode. Payload data as well as transmit and receive status are passed on these busses.

A third internal bus is used to access the MAC's "Control and Status Registers" (CSR's). This bus is also accessible from the host.

On the backend, the MAC interfaces with the 10/100 PHY through an MII (Media Independent Interface) port which is internal to the LAN9220. The MAC CSR's also provide a mechanism for accessing the PHY's internal registers through the internal SMI (Serial Management Interface) bus.

The receive and transmit FIFOs allow increased packet buffer storage to the MAC. The FIFOs are a conduit between the host interface and the MAC through which all transmitted and received data and status information is passed. Deep FIFOs allow a high degree of latency tolerance relative to the various transport and OS software stacks reducing and minimizing overrun conditions. Like the MAC, the FIFOs have separate receive and transmit data paths.

The LAN9220 can store up to 250 Ethernet packets utilizing FIFOs, totaling 16K bytes, with a packet granularity of 4 bytes. This memory is shared by the RX and TX blocks and is configurable in terms of allocation. This depth of buffer storage minimizes or eliminates receive overruns.

3.2 Flow Control

The LAN9220 Ethernet MAC supports full-duplex flow control using the pause operation and control frame. It also supports half-duplex flow control using back pressure.

3.2.1 FULL-DUPLEX FLOW CONTROL

The pause operation inhibits data transmission of data frames for a specified period of time. A Pause operation consists of a frame containing the globally assigned multicast address (01-80-C2-00-00-01), the PAUSE opcode, and a parameter indicating the quantum of slot time (512 bit times) to inhibit data transmissions. The PAUSE parameter may range from 0 to 65,535 slot times. The Ethernet MAC logic, on receiving a frame with the reserved multicast address and PAUSE opcode, inhibits data frame transmissions for the length of time indicated. If a Pause request is received while a transmission is in progress, then the pause will take effect after the transmission is complete. Control frames are received and processed by the MAC and are passed on.

The MAC also transmits control frames (pause command) via both hardware and software control. The software driver requests the MAC to transmit a control frame and gives the value of the PAUSE time to be used in the control frame. The MAC Function constructs a control frame with the appropriate values set in all the different fields (as defined in the 802.3x specification) and transmits the frame to the MII interface. The transmission of the control frame is not affected by the current state of the Pause timer value that is set because of a recently received control frame.

3.2.2 HALF-DUPLEX FLOW CONTROL (BACKPRESSURE)

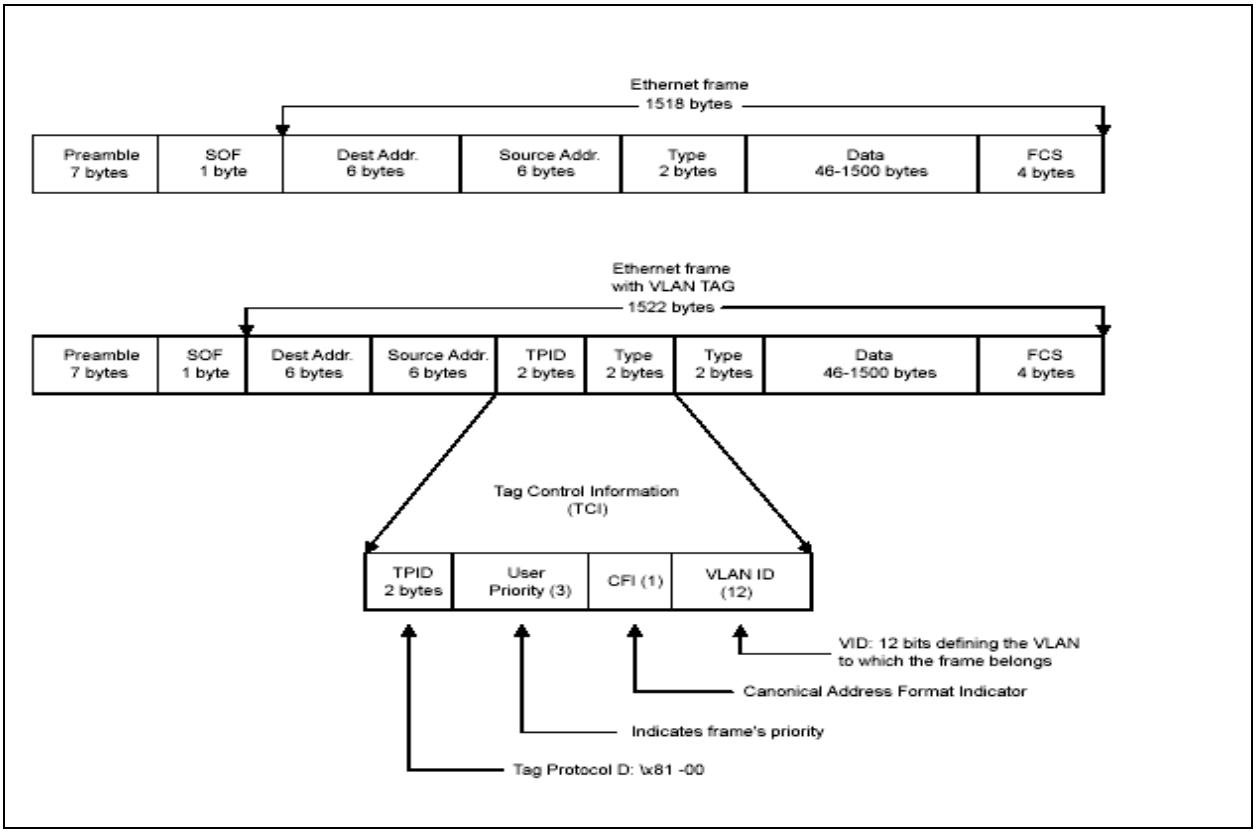
In half-duplex mode, back pressure is used for flow control. Whenever the receive buffer/FIFO becomes full or crosses a certain threshold level, the MAC starts sending a Jam signal. The MAC transmit logic enters a state at the end of current transmission (if any), where it waits for the beginning of a received frame. Once a new frame starts, the MAC starts sending the jam signal, which will result in a collision. After sensing the collision, the remote station will back off its transmission. The MAC continues sending the jam to make other stations defer transmission. The MAC only generates this collision-based back pressure when it receives a new frame, in order to avoid any late collisions.

3.2.3 VIRTUAL LOCAL AREA NETWORK (VLAN) SUPPORT

Virtual Local Area Networks or VLANs, as defined within the IEEE 802.3 standard, provide network administrators one means of grouping nodes within a larger network into broadcast domains. To implement a VLAN, four extra bytes are added to the basic Ethernet packet. As shown in Figure 3-1, "VLAN Frame", the four bytes are inserted after the Source Address Field and before the Type/Length field. The first two bytes of the VLAN tag identify the tag, and by convention are set to the value 0x8100. The last two bytes identify the specific VLAN associated with the packet; they also provide a priority field.

The LAN9220 supports VLAN-tagged packets. The LAN9220 provides two registers which are used to identify VLAN-tagged packets. One register should normally be set to the conventional VLAN ID of 0x8100. The other register provides a way of identifying VLAN frames tagged with a proprietary (not 0x8100) identifier. If a packet arrives bearing either of these tags in the two bytes succeeding the Source Address field, the controller will recognize the packet as a VLAN-tagged packet. In this case, the controller increases the maximum allowed packet size from 1518 to 1522 bytes (normally the controller filters packets larger than 1518 bytes). This allows the packet to be received, and then processed by host software, or to be transmitted on the network.

FIGURE 3-1: VLAN FRAME



3.3 Address Filtering Functional Description

The Ethernet address fields of an Ethernet Packet, consists of two 6-byte fields: one for the destination address and one for the source address. The first bit of the destination address signifies whether it is a physical address or a multicast address.

The LAN9220 address check logic filters the frame based on the Ethernet receive filter mode that has been enabled. Filter modes are specified based on the state of the control bits in Table 3-1, "Address Filtering Modes", which shows the various filtering modes used by the Ethernet MAC Function. These bits are defined in more detail in the “MAC Control Register”. Please refer to Section 5.4.1, "MAC_CR—MAC Control Register" for more information on this register.

If the frame fails the filter, the Ethernet MAC function does not receive the packet. The host has the option of accepting or ignoring the packet.

TABLE 3-1: ADDRESS FILTERING MODES

MCPAS	PRMS	INVFILT	HO	HPFILT	Description
0	0	0	0	0	MAC address perfect filtering only for all addresses.
0	0	0	0	1	MAC address perfect filtering for physical address and hash filtering for multicast addresses
0	0	0	1	1	Hash Filtering for physical and multicast addresses
0	0	1	0	0	Inverse Filtering

TABLE 3-1: ADDRESS FILTERING MODES (CONTINUED)

MCPAS	PRMS	INVFILT	HO	HPFILT	Description
X	1	0	X	X	Promiscuous
1	0	0	0	X	Pass all multicast frames. Frames with physical addresses are perfect-filtered
1	0	0	1	1	Pass all multicast frames. Frames with physical addresses are hash-filtered

3.4 Filtering Modes

3.4.1 PERFECT FILTERING

This filtering mode passes only incoming frames whose destination address field exactly matches the value programmed into the MAC Address High register and the MAC address low register. The MAC address is formed by the concatenation of the above two registers in the MAC CSR Function.

3.4.2 HASH ONLY FILTERING

This type of filtering checks for incoming Receive packets with either multicast or physical destination addresses, and executes an imperfect address filtering against the hash table.

During imperfect hash filtering, the destination address in the incoming frame is passed through the CRC logic and the upper six bits of the CRC register are used to index the contents of the hash table. The hash table is formed by merging the register's multicast hash table high and multicast hash table low in the MAC CSR Function to form a 64-bit hash table. The most significant bit determines the register to be used (High/Low), while the other five bits determine the bit within the register. A value of 00000 selects Bit 0 of the multicast hash table low register and a value of 11111 selects Bit 31 of the multicast hash table high register.

3.4.2.1 Hash Perfect Filtering

In hash perfect filtering, if the received frame is a physical address, the LAN9220 Packet Filter block perfect-filters the incoming frame's destination field with the value programmed into the MAC Address High register and the MAC Address Low register. If the incoming frame is a multicast frame, however, the LAN9220 packet filter function performs an imperfect address filtering against the hash table.

The imperfect filtering against the hash table is the same imperfect filtering process described in the "Hash Only Filtering" section above.

3.4.2.2 Inverse Filtering

In inverse filtering, the Packet Filter Block accepts incoming frames with a destination address not matching the perfect address (i.e., the value programmed into the MAC Address High register and the MAC Address Low register in the CRC block and rejects frames with destination addresses matching the perfect address).

For all filtering modes, when the MCPAS bit is set, all multicast frames are accepted. When the PRMS bit is set, all frames are accepted regardless of their destination address. This includes all broadcast frames as well.

3.5 Wake-up Frame Detection

Setting the Wake-Up Frame Enable bit (WUEN) in the "WUCSR—Wake-up Control and Status Register", places the LAN9220 MAC in the wake-up frame detection mode. In this mode, normal data reception is disabled, and detection logic within the MAC examines receive data for the pre-programmed wake-up frame patterns. The LAN9220 can be programmed to notify the host of the wake-up frame detection with the assertion of the host interrupt (IRQ) or assertion of the power management event signal (PME). Upon detection, the Wake-Up Frame Received bit (WUFR) in the WUCSR is set. When the host clears the WUEN bit the LAN9220 will resume normal receive operation.

Before putting the MAC into the wake-up frame detection state, the host must provide the detection logic with a list of sample frames and their corresponding byte masks. This information is written into the Wake-up Frame Filter register (WUFF). Please refer to Section 5.4.11, "WUFF—Wake-up Frame Filter," on page 102 for additional information on this register.

The MAC supports four programmable filters that support many different receive packet patterns. If remote wake-up mode is enabled, the remote wake-up function receives all frames addressed to the MAC. It then checks each frame against the enabled filter and recognizes the frame as a remote wake-up frame if it passes the wakeup frame filter register's address filtering and CRC value match.

In order to determine which bytes of the frames should be checked by the CRC module, the MAC uses a programmable byte mask and a programmable pattern offset for each of the four supported filters.

The pattern's offset defines the location of the first byte that should be checked in the frame. Since the destination address is checked by the address filtering Function, the pattern offset is always greater than 12.

The byte mask is a 31-bit field that specifies whether or not each of the 31 contiguous bytes within the frame, beginning in the pattern offset, should be checked. If bit j in the byte mask is set, the detection logic checks byte offset +j in the frame. In order to load the Wake-up Frame Filter register, the host LAN driver software must perform eight writes to the Wake-up Frame Filter register (WUFF). The Diagram shown in Table 3-2, "Wake-Up Frame Filter Register Structure" below, shows the wake-up frame filter register's structure.

Note 3-1 Wake-up frame detection can be performed when the LAN9220 is in the D0 or D1 power states. In the D0 state, wake-up frame detection is enabled when the WUEN bit is set.

Note 3-2 Wake-up frame detection, as well as Magic Packet detection, is always enabled and cannot be disabled when the device enters the D1 state.

Note 3-3 When wake-up frame detection is enabled via the WUEN bit of the WUCSR—Wake-up Control and Status Register, a broadcast wake-up frame will wake-up the device despite the state of the Disable Broadcast Frame (BCAST) bit in the MAC_CR—MAC Control Register.

TABLE 3-2: WAKE-UP FRAME FILTER REGISTER STRUCTURE

Filter 0 Byte Mask							
Filter 1 Byte Mask							
Filter 2 Byte Mask							
Filter 3 Byte Mask							
Reserved	Filter 3 Command	Reserved	Filter 2 Command	Reserved	Filter 1 Command	Reserved	Filter 0 Command
Filter 3 Offset		Filter 2 Offset		Filter 1Offset		Filter 0 Offset	
Filter 1 CRC-16				Filter 0 CRC-16			
Filter 3 CRC-16				Filter 2 CRC-16			

The Filter i Byte Mask defines which incoming frame bytes Filter i will examine to determine whether or not this is a wake-up frame. Table 3-3, describes the byte mask's bit fields.

TABLE 3-3: FILTER I BYTE MASK BIT DEFINITIONS

Filter i Byte Mask Description	
Field	**Description**
31	Must be zero (0)
30:0	**Byte Mask:** If bit j of the byte mask is set, the CRC machine processes byte number pattern - (offset + j) of the incoming frame. Otherwise, byte pattern - (offset + j) is ignored.

The Filter i command register controls Filter i operation. Table 3-4 shows the Filter I command register.

TABLE 3-4: FILTER I COMMAND BIT DEFINITIONS

Filter i Commands	
Field	**Description**
3	**Address Type:** Defines the destination address type of the pattern. When bit is set, the pattern applies only to multicast frames. When bit is cleared, the pattern applies only to unicast frames.
2:1	**RESERVED**
0	**Enable Filter:** When bit is set, Filter i is enabled, otherwise, Filter i is disabled.

The Filter i Offset register defines the offset in the frame's destination address field from which the frames are examined by Filter i. Table 3-5 describes the Filter i Offset bit fields.

TABLE 3-5: FILTER I OFFSET BIT DEFINITIONS

Filter i Offset Description	
Field	**Description**
7:0	**Pattern Offset:** The offset of the first byte in the frame on which CRC is checked for wake-up frame recognition. The minimum value of this field must be 12 since there should be no CRC check for the destination address and the source address fields. The MAC checks the first offset byte of the frame for CRC and checks to determine whether the frame is a wake-up frame. Offset 0 is the first byte of the incoming frame's destination address.

The Filter i CRC-16 register contains the CRC-16 result of the frame that should pass Filter i.

Table 3-6 describes the Filter i CRC-16 bit fields.

TABLE 3-6: FILTER I CRC-16 BIT DEFINITIONS

Filter i CRC-16 Description	
Field	**Description**
15:0	**Pattern CRC-16:** This field contains the 16-bit CRC value from the pattern and the byte mask programmed to the wake-up filter register Function. This value is compared against the CRC calculated on the incoming frame, and a match indicates the reception of a wakeup frame.

3.5.1 MAGIC PACKET DETECTION

Setting the Magic Packet Enable bit (MPEN) in the "WUCSR—Wake-up Control and Status Register", places the LAN9220 MAC in the "Magic Packet" detection mode. In this mode, normal data reception is disabled, and detection logic within the MAC examines receive data for a Magic Packet. The LAN9220 can be programmed to notify the host of the "Magic Packet" detection with the assertion of the host interrupt (IRQ) or assertion of the power management event signal (PME). Upon detection, the Magic Packet Received bit (MPR) in the WUCSR is set. When the host clears the MPEN bit the LAN9220 will resume normal receive operation. Please refer to Section 5.4.12, "WUCSR—Wake-up Control and Status Register," on page 102 for additional information on this register.

In Magic Packet mode, the Power Management Logic constantly monitors each frame addressed to the node for a specific Magic Packet pattern. It checks only packets with the MAC's address or a broadcast address to meet the Magic Packet requirement. The Power Management Logic checks each received frame for the pattern 48h FF_FF_FF_FF_FF_FF after the destination and source address field.

Then the Function looks in the frame for 16 repetitions of the MAC address without any breaks or interruptions. In case of a break in the 16 address repetitions, the PMT Function scans for the 48'hFF_FF_FF_FF_FF_FF pattern again in the incoming frame.

The 16 repetitions may be anywhere in the frame but must be preceded by the synchronization stream. The device will also accept a multicast frame, as long as it detects the 16 duplications of the MAC address. If the MAC address of a node is 00h 11h 22h 33h 44h 55h, then the MAC scans for the following data sequence in an Ethernet: Frame.

Destination Address Source Address ……FF FF FF FF FF FF

00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55

00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55

00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55

00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55

…CRC

It should be noted that Magic Packet detection can be performed when LAN9220 is in the D0 or D1 power states. In the D0 state, "Magic Packet" detection is enabled when the MPEN bit is set. In the D1 state, Magic Packet detection, as well as wake-up frame detection, are automatically enabled when the device enters the D1 state.

3.6 Checksum Offload Engines (COE)

The LAN9220 contains two checksum offload engines, which offload the calculation of the 16-bit checksum for transmitted and received Ethernet frames. The functionality of the checksum offload engines is described in the following sections:

- Receive Checksum Offload Engine (RXCOE)
- Transmit Checksum Offload Engine (TXCOE)

3.6.1 RECEIVE CHECKSUM OFFLOAD ENGINE (RXCOE)

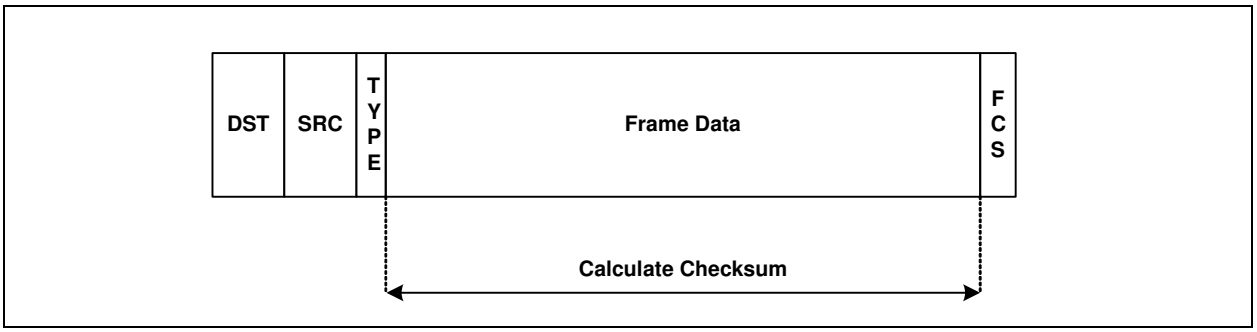
The receive checksum offload engine provides assistance to the CPU by calculating a 16-bit checksum for a received Ethernet frame. The RXCOE readily supports the following IEEE802.3 frame formats:

- Type II Ethernet frames
- SNAP encapsulated frames
- Support for up to 2, 802.1q VLAN tags

The resulting checksum value can also be modified by software to support other frame formats.

The RXCOE has two modes of operation. In mode 0, the RXCOE calculates the checksum between the first 14 bytes of the Ethernet frame and the FCS. This is illustrated in Figure 3-2.

FIGURE 3-2: RXCOE CHECKSUM CALCULATION



In mode 1, the RXCOE supports VLAN tags and a SNAP header. In this mode the RXCOE calculates the checksum at the start of L3 packet. The VLAN1 tag register is used by the RXCOE to indicate what protocol type is to be used to indicate the existence of a VLAN tag. This value is typically 8100h.

Example frame configurations:

FIGURE 3-3: TYPE II ETHERNET FRAME

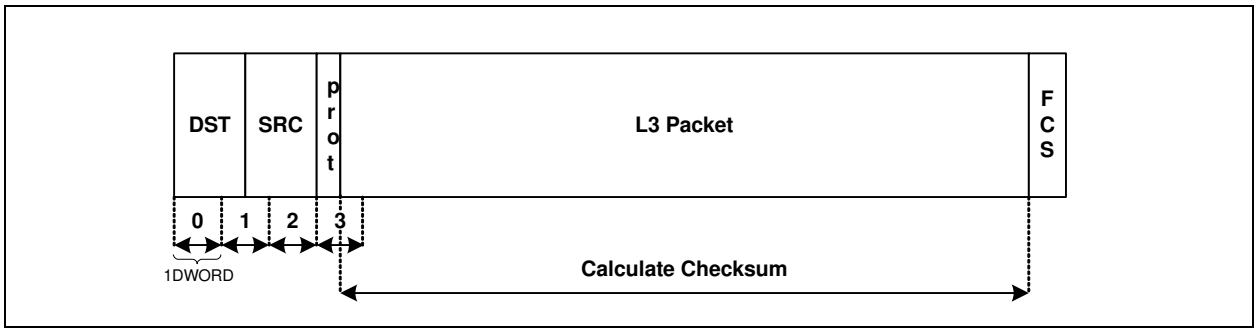


FIGURE 3-4: ETHERNET FRAME WITH VLAN TAG

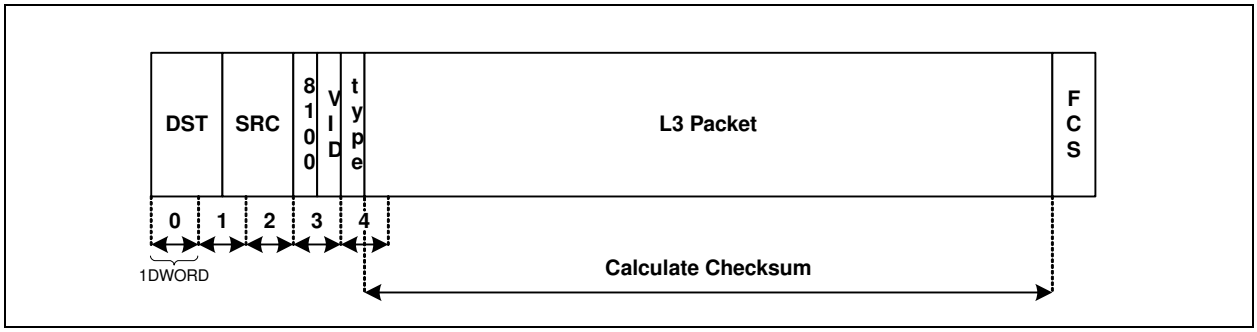


FIGURE 3-5: ETHERNET FRAME WITH LENGTH FIELD AND SNAP HEADER

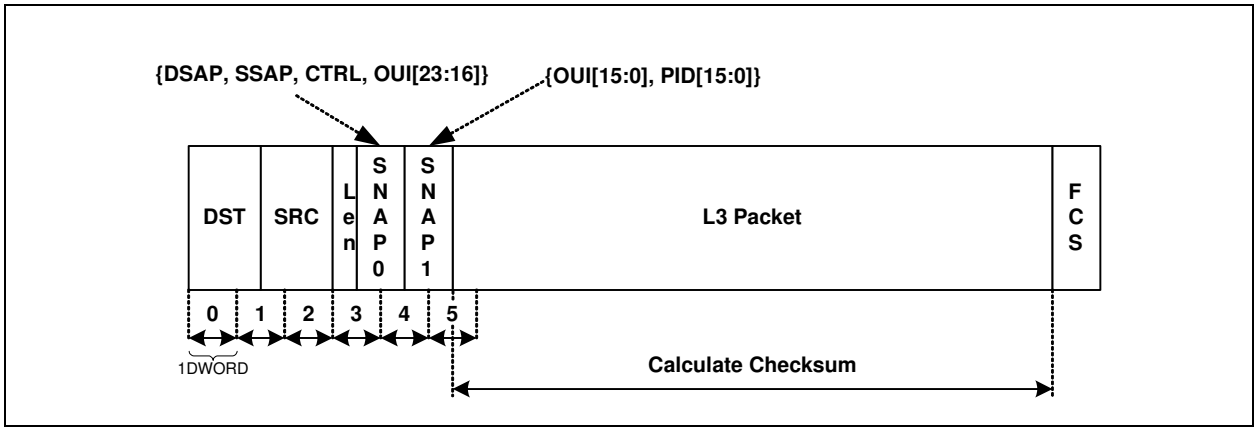


FIGURE 3-6: ETHERNET FRAME WITH VLAN TAG AND SNAP HEADER

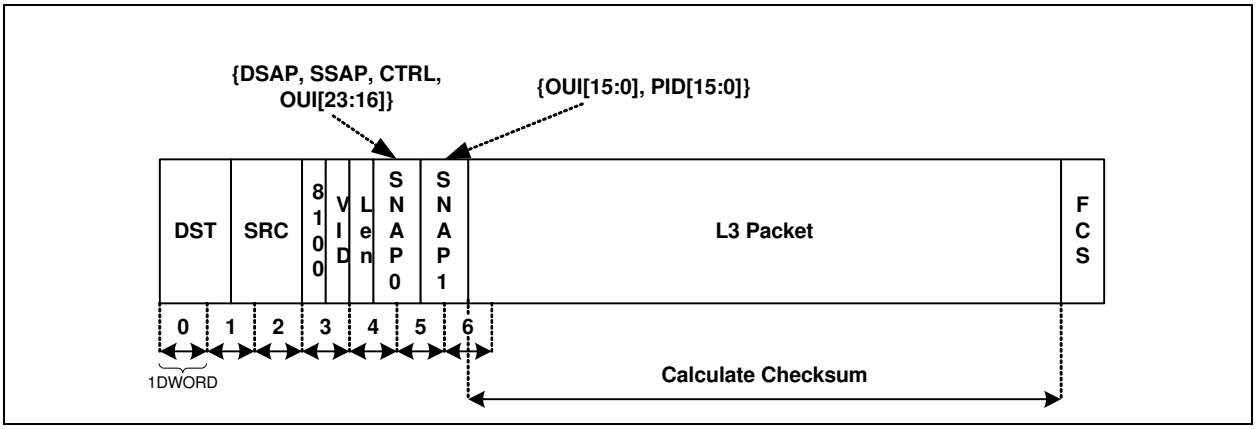
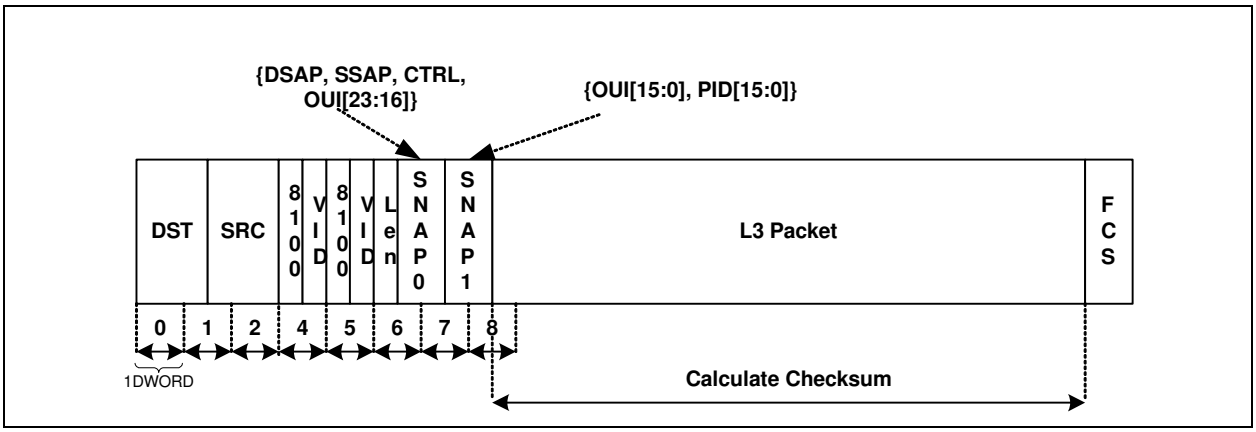


FIGURE 3-7: ETHERNET FRAME WITH MULTIPLE VLAN TAGS AND SNAP HEADER



The RXCOE supports a maximum of two VLAN tags. If there are more than two VLAN tags, the VLAN protocol identifier for the third tag is treated as an Ethernet type field. The checksum calculation will begin immediately after the type field.

The RXCOE resides in the RX path within the MAC. As the RXCOE receives an Ethernet frame it calculates the 16-bit checksum. The RXCOE passes the Ethernet frame to the RX Data FIFO with the checksum appended to the end of the frame. The RXCOE inserts the checksum immediately after the last byte of the Ethernet frame. The packet length field in the RX status word (refer to Section 3.13.3) will indicate that the frame size has increased by two bytes to accommodate the checksum.

Setting the RXCOE_EN bit in the COE_CR—Checksum Offload Engine Control Register enables the RXCOE, while the RXCOE_MODE bit selects the operating mode. When the RXCOE is disabled, the received data is simply passed through the RXCOE unmodified.

Note:

- Software applications must stop the receiver and flush the RX data path before changing the state of the RXCOE_EN or RXCOE_MODE bits.
- When the RXCOE is enabled, automatic pad stripping must be disabled (bit 8 (PADSTR) of the MAC_CR—MAC Control Register) and vice versa. These functions cannot be enabled simultaneously.

3.6.2 RX CHECKSUM CALCULATION

The checksum is calculated 16 bits at a time. In the case of an odd sized frame, an extra byte of zero is used to pad up to 16 bits.

Consider the following packet: DA, SA, Type, B0, B1, B2 … BN, FCS

Let [A, B] = A*256 + B;

If the packet has an even number of octets then

checksum = [B1, B0] + C0 + [B3, B2] + C1 + … + [BN, BN-1] + CN-1

Where C0, C1, ... CN-1 are the carry out results of the intermediate sums.

If the packet has an odd number of octets then

checksum = [B1, B0] + C0 + [B3, B2] + C1 + … + [0, BN] + CN-1

3.6.3 TRANSMIT CHECKSUM OFFLOAD ENGINE (TXCOE)

The transmit checksum offload engine provides assistance to the CPU by calculating a 16-bit checksum, typically for TCP, for a transmit Ethernet frame. The TXCOE calculates the checksum and inserts the results back into the data stream as it is transferred to the MAC.

To activate the TXCOE and perform a checksum calculation, the host must first set the TX checksum offload engine enable bit (TXCOE_EN) in the COE_CR—Checksum Offload Engine Control Register. The host then pre-pends a 3 DWORD buffer to the data that will be transmitted. The pre-pended buffer includes a TX Command ‘A’, TX Command ‘B’, and a 32-bit TX checksum preamble. When bit 14 (CK) of the TX Command ‘B’ is set in conjunction with bit 13 (FS) of TX Command ‘A’ and bit 16 (TXCOE_EN) of the COE_CR register, the TXCOE will perform a checksum calculation on the associated packet. When these three bits are set, a 32-bit TX checksum preamble must be pre-pended to the beginning of the TX packet (refer to Table 3-7). The TX checksum preamble instructs the TXCOE on the handling of the associated packet. Bits 11:0 of the TX checksum preamble define the byte offset at which the data checksum calculation will begin (TXCSSP). The checksum calculation will begin at this offset and will continue until the end of the packet. The data checksum calculation must not begin in the MAC header (first 14 bytes) or in the last 4 bytes of the TX packet. When the calculation is complete, the checksum will be inserted into the packet at the byte offset defined by bits 27:16 of the TX checksum preamble (TXCSLOC). The TX checksum cannot be inserted in the MAC header (first 14 bytes) or in the last 4 bytes of the TX packet. If the CK bit is not set in the first TX Command ‘B’ of a packet, the packet is passed directly through the TXCOE without modification, regardless if the TXCOE_EN is set. An example of a TX packet with a pre-pended TX checksum preamble can be found in Section 3.12.6.3, "TX Example 3". In this example the host writes the packet data to the ethernet controller in four fragments, the first containing the TX Checksum Preamble. Figure 3-23 shows how these fragments are loaded into the TX Data FIFO. For more information on the TX Command ‘A’ and TX Command ‘B’, refer to Section 3.12.2, "TX Command Format".

If the TX packet already includes a partial checksum calculation (perhaps inserted by an upper layer protocol), this checksum can be included in the hardware checksum calculation by setting the TXCSSP field in the TX checksum preamble to include the partial checksum. The partial checksum can be replaced by the completed checksum calculation by setting the TXCSLOC pointer to point to the location of the partial checksum.

TABLE 3-7: TX CHECKSUM PREAMBLE

Field	Description
31:28	**RESERVED**
27:16	**TXCSLOC - TX Checksum Location** This field specifies the byte offset where the TX checksum will be inserted in the TX packet. The checksum will replace two bytes of data starting at this offset. **Note:** The TX checksum cannot be inserted in the MAC header (first 14 bytes) or in the last 4 bytes of the TX packet.
15:12	**RESERVED**
11:0	**TXCSSP - TX Checksum Start Pointer** This field indicates start offset, in bytes, where the checksum calculation will begin in the associated TX packet. **Note:** The data checksum calculation must not begin in the MAC header (first 14 bytes) or in the last 4 bytes of the TX packet.

Note:

- When the TXCOE is enabled, the third DWORD of the pre-pended packet is not transmitted. However, 4 bytes must be added to the packet length field in TX Command 'B'.
- The TX checksum preamble must be DWORD-aligned (i.e., the two least significant bits of the Data Start Offset fields in TX Command "A" must be zero). Any valid buffer end alignment setting can be used.
- Software applications must stop the transmitter and flush the TX data path before changing the state of the TXCOE_EN bit. However, the CK bit of TX Command 'B' can be set or cleared on a per-packet basis.

3.6.3.1 TX Checksum Calculation

The TX checksum calculation is performed using the same operation as the RX checksum shown in Section 3.6.2, with the exception that the calculation starts as indicated by the preamble, and the transmitted checksum is the one's-compliment of the final calculation.

Note: When the TX checksum offload feature is invoked, if the calculated checksum is 0000h, it is left unaltered. UDP checksums are optional under IPv4, and a zero checksum calculated by the TX checksum offload feature will erroneously indicate to the receiver that no checksum was calculated, however, the packet will typically not be rejected by the receiver. Under IPv6, however, according to RFC 2460, the UDP checksum is not optional. A calculated checksum that yields a result of zero must be changed to FFFFh for insertion into the UDP header. IPv6 receivers discard UDP packets containing a zero checksum. **Thus, this feature must not be used for UDP checksum calculation under IPv6.**