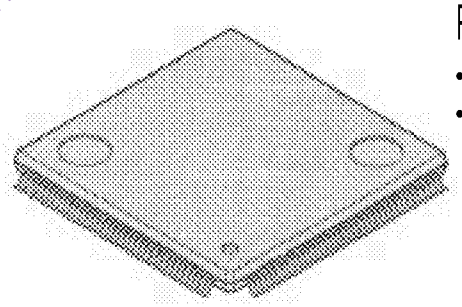
PAUSE FLOW CONTROL

10Mb/s-/100Mb/s Ethernet Controller

MB86974



Package

- 144-pin, plastic QFP
- FPT-144P-M08

Description

The Fujitsu MB86974 is a high-quality Ethernet controller that offers many benefits and advantages to its users.

The controller operates at either 100-Mbit/s or 10-Mbit/s, and fully supports the IEEE 802.3u flow control.

When in half-duplex mode, the controller implements the Institute of Electrical and Electronics Engineers (IEEE) 802.3 Carrier Sense Multiple Access with Collision Detection (CSMA/CD) protocol. When in full-duplex mode, it implements the IEEE 802.3 Media Access Controller (MAC) control layer.

The controller also offers the following: (1) PAUSE operation for flow control, including programmable support for additional MAC control functions; (2) direct connection to the 32-bit Peripheral Components Interconnect (PCI) local bus, using the bus-master burst transfer mode to efficiently move data to and from system memory; (3) on-chip memory for buffering, so external local-buffer memory is not needed; (4) an optional 10 Mb/s 7-wire interface.

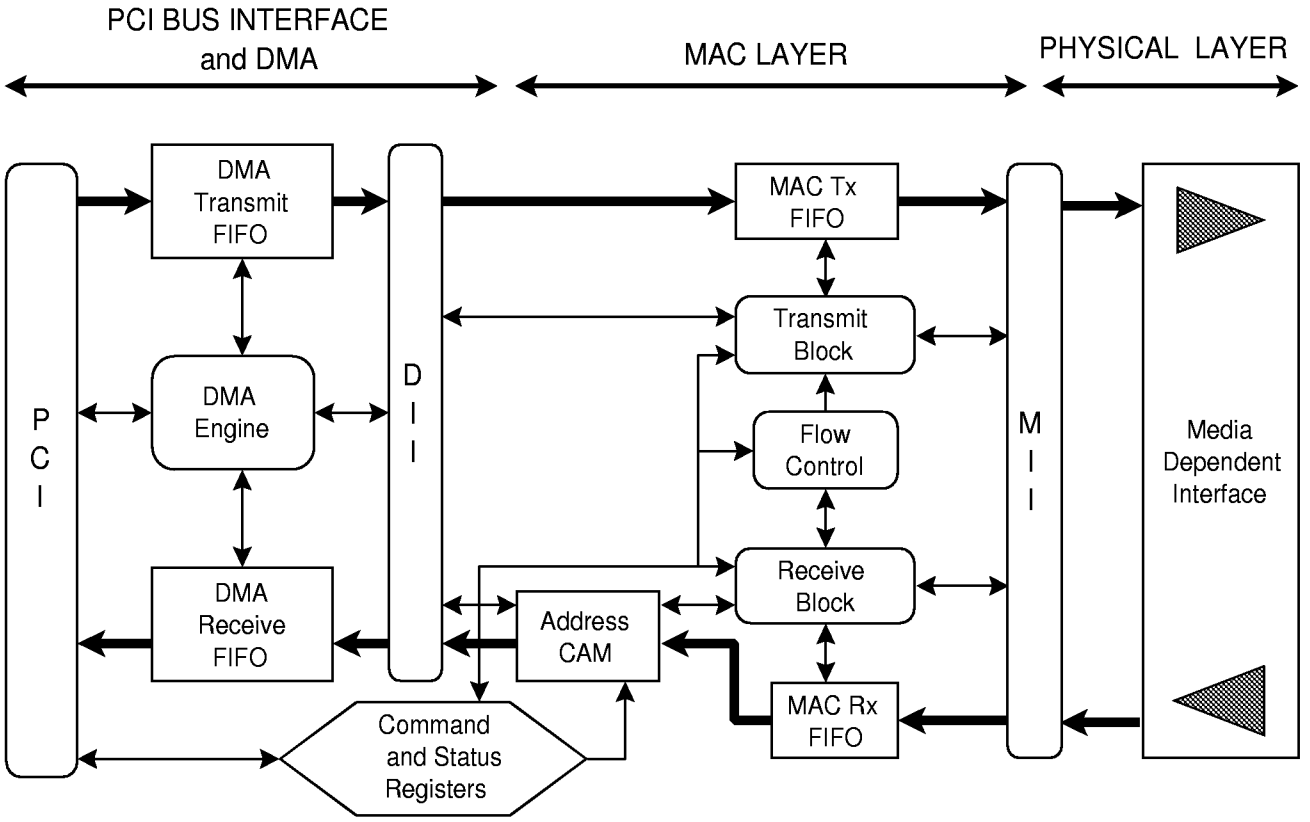
Features

- 100/10-Mbit/s operation, full Media Independent Interface (MII)
- Low-power Complementary Metal Oxide Semiconductor (CMOS) design, 3.3-volt operation (5V tolerant I/O)
- Direct PCI interface with DMA burst mode, burst size control, big- and little- endian support
- IEEE 802.3u standard compliance
- Flow-control PAUSE operation, power management, early transmit and receive
- Large arbitrated Direct Memory Access (DMA) and First-In, First-Out (FIFO) buffers
- Full-duplex capability
- On-chip Content Addressable Memory (CAM) and optional external CAM
- Electrically Erasable Programable Read Only Memory (EEPROM) support
- 144-pin Plastic Quad Flat Package (PQFP)

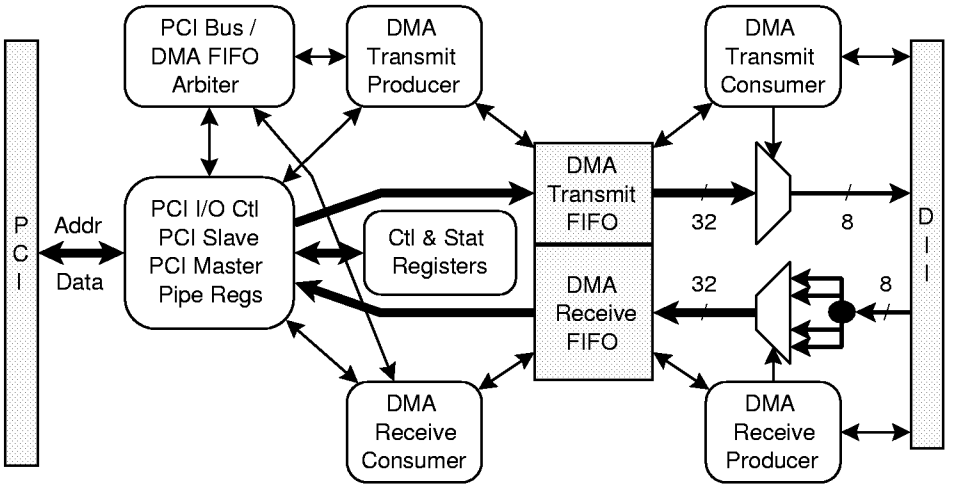
Preliminary 12/14/98

Table of Contents

MB86974 Ethernet Controller Block Diagram



PCI DMA Functional Blocks



PCI and DMA Blocks

The PCI bus is an industry-standard bus for high-performance PCs and workstations. Some network hub designs use the PCI bus for internal data paths, or for backplane connections. The speed of the PCI bus allows it to support 100-Mbit/s Ethernet. The bus is cost effective. In bus-master mode, the bus relieves the system of significant processing load.

The PCI bus features 32-bit addresses and a 32-bit data path, multiplexed over the same pins. In burst-mode transfers, the PCI bus can, in theory, support 132-Mbyte/s transfer rates with a 33-MHz system clock. The MB86974 Flow Control 100/10-Mbit/s Ethernet controller works with both current and future higher-performance bus-controller chips.

The diagram above shows the major functional blocks in the DMA for configuring and controlling the PCI bus, and for transferring data to and from the DMA FIFOs.

PCI and DMA Overview

The PCI I/O control block generates and recognizes PCI control signals. The PCI slave block recognizes and controls transactions where the Ethernet controller is the target device. The PCI master block controls transactions in which the Ethernet controller is the initiator device. The PCI pipe registers provide buffering, so the DMA engine can sustain back-to-back performance during long burst operations.

The control and status registers are for PCI configuration and MAC and DMA control. An arbiter block allocates access to the PCI bus and DMA memory.

The DMA engine controls two FIFO buffers. The *DMA Transmit* FIFO holds data and status information for packets being transmitted. The *DMA Receive* FIFO holds data and status information for packets being received.

Each FIFO has a producer control block, controling data being placed in the FIFO, and a consumer control block that controls data being taken from the FIFO. The logically distinct FIFOs are realized within a single memory block, but this is transparent to the user.

The PCI bus/DMA FIFO arbiter decides which of the consumer and producer state machines has highest priority for accessing the PCI bus and the FIFO buffers. The priority is dynamic. During burst transfers, the state machines controlling the PCI bus are given priority, if possible. But if a receive FIFO is close to becoming full, or if a transmit FIFO is close to becoming empty during transmission, that FIFO receives priority. Otherwise, the arbiter provides round-robin fair service.

The DMA controller blocks provide logic for controlling bus-master read and write operations across the PCI bus. They include:

- Burst size control to optimize PCI and system performance
- Transmit threshold control to match transmission latency to PCI bus latency
- Big and little endian byte swapping to support transfering of data to big endian or little endian processors
- Buffer spanning and packing controls
- Polling controls to optionally poll for packets to transmit
- Wake-up controls to start transmission as soon as data is ready
- Early notification to allow processing of incoming data to begin before data end
- Interrupt enable controls to adjust controller behavior to protocol requirements

The registers that control the PCI are listed on page 12.

PCI Arbiter

The PCI arbiter accepts as input status information from the DMA transmit and receive controllers. The arbiter controls switching among transmit, receive, and notification activities on the PCI bus using PCI master-mode access. The arbiter also allows slave-mode access to read and write control and status registers.

DMA Transmit Controller

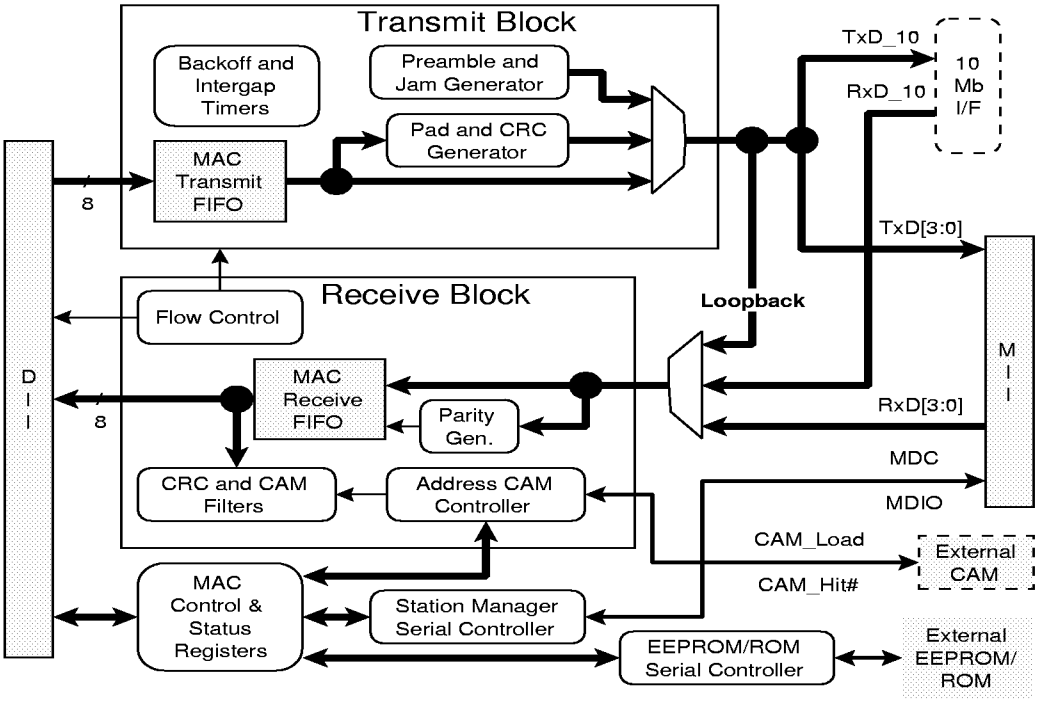
The DMA transmit controller consists of two state machines; a producer and a consumer. The *DMA transmit producer* reads the transmit-queue frame descriptor and controls the transfer of data from the transmit-queue to the transmit data FIFO. It also controls the writing of transmit status information, which gives MAC status information about transmitted packets when transmission is complete. The *DMA transmit consumer* controls the MAC transmit engine and the movement of data from the DMA transmit FIFO to the smaller MAC transmit FIFO.

DMA Receive Controller

The DMA receive controller consists of two state machines: a producer and a consumer. The *DMA receive producer* controls the movement of data from the MAC to the DMA receive FIFO; and the *DMA receive consumer* allocates buffers from the free buffer list, writes new frame descriptors and related buffer descriptors in the free descriptor area, and controls the transfer of bytes from the DMA receive FIFO into system memory via the PCI bus.

The registers that control the DMA are listed on page 16.

MAC and MII Blocks



MAC Functional Overview

The MAC consists of a transmit block, a receive block, and a set of control and status registers. It also has two serial controllers, one for accessing the Media Independent Interface (MII) station management interface, and one for the external EEPROM/ROM. The MAC has a loopback circuit and also provides optional support for a 10-Mbit/s interface to an external Manchester encoder/decoder/transceiver. In addition, it supports an optional external CAM circuit.

The MII is the interface between a 100 BASE-T-compatible physical layer and the transmit and receive blocks. It is part of the IEEE 802.3u standard. The transmit block buffers the outgoing data in the MAC transmit FIFO, encapsulates it and passes it on to the MII or the 10 Mb/s interface.

Additionally, the transmit block has circuits for generating preamble, jam and pad bytes, and for generating the Cyclic Redundancy Check (CRC) value. It has logic to check parity and a timer for the backoff delay after a collision. The block also has a timer for the inter-packet gap after a transmission.

The receive block decapsulates the received packet from the MII or the 10 Mb/s interface and stores it in the MAC receive FIFO. The receive block checks the CRC value, generates parity to protect data in the FIFO, and checks packet lengths. The receive block also has a CAM block that provides for acceptance or rejection of a packet based on its destination address.

The registers control programmable options and specify which conditions interrupt the system. The status registers hold information for error handling software. The error counters accumulate statistical information for network management software.

The loopback circuit provides for MAC-layer testing in isolation from the MII and physical layer.

The MAC blocks provide controls for network operation that include:

- Controls to enable and disable the transmit and receive circuits, including requests to halt at the end of the current packet
- Interrupt enable and disable controls for individual conditions

- Address recognition controls for up to 21 individual addresses
- Counters and status bits for collecting network management data
- Loopback and other controls to aid in diagnosing network problems
- Controls for reading, writing, and erasing values in an EEPROM or ROM chip
- PAUSE operation enable controls to cause the transmitter to pause on receipt of MAC control packets specifying a PAUSE operation
- MAC control packet transmit controls to enable generation of PAUSE and other MAC control packets, even when the transmitter is paused
- MAC control packet pass-through controls to enable software to process other forms of MAC control packets

MAC Transmit Block

The transmit block complies with the IEEE 802.3 standard for Carrier Sense Multiple Access with Collision Detection (CSMA/CD) protocol. It also supports the standard full-duplex mode of operation, which allows simultaneous transmission and reception.

The transmit block consists of the Transmit FIFO, FIFO controller and counters, the preamble, jam, pad byte and CRC generators, parity checker, and backoff and intergap timers.

MAC Receive Block

The receive block complies with the IEEE 802.3 standard for CSMA/CD protocol. It also supports the full-duplex mode of operation that allows simultaneous transmission and reception.

The receive block consists of the Receive FIFO, FIFO controller and counters, the CAM block for address recognition, the CRC generator and checker, and the parity generator.

Flow Control Block

The flow control block provides the following functions: recognition of MAC control frames received by the receive block; transmission of MAC control frames, even if the transmitter is paused; timers and counters for PAUSE operation; Command and Status Register (CSR) interface; and options for passing MAC control frames through to software drivers.

The receive logic in the flow-control block recognizes a MAC control frame and performs the PAUSE operation after verifying the following conditions:

- The length/type field has the special value specified for MAC control frames
- The CAM recognizes the destination address
- The frame length is 64 bytes, including CRC
- That CRC is good
- The frame contains a valid PAUSE operation code and operand value

If the CAM does not recognize the destination address, the MAC rejects the packet. If the packet length is not 64 bytes, including CRC, or if the length/type field does not have the special value specified for MAC control frames, the MAC takes no special action and the packet is treated as normal . If the packet passes all these checks, it is marked as a MAC control packet and acted upon. It may also be passed forward to the software drivers.

A control bit in the Transmit Status Register can be set to generate a full-duplex PAUSE operation or other MAC control function, even if the transmitter itself is paused.

Two timers are used during PAUSE operations. Each timer has a corresponding CSR register. The Received Pause Count Register and timer pair is used when a received packet causes the transmitter to pause. The Remote Pause Count pair is used to approximate the pause status of the other end of the link after the transmitter sends a PAUSE command.

The Command and Status Register (CSR) interface includes control and status bits within the transmit and receive control registers and status registers that allow: (1) sending a MAC control frame; (2) enabling and disabling MAC control functions; and (3) reading the flow control counters.

Control bits are provided for either processing MAC control frames entirely within the controller, or for passing on MAC control frames to the software drivers. This allows a higher level of software to handle flow control operations. The flow control registers are described in more detail on page 21.

MAC Control and Status Registers

The MAC has a block of registers that provides control and status information. These registers control the transmit and receive blocks and report MAC status. They also provide a communication interface to the CAM, the MII station management interface, and the optional external EEPROM/ROM. These registers are available via PCI memory-mapped or I/O-mapped access. For more information on the MAC control and status registers, see the table on page 22.

Media Independent Interface (MII)

The MII provides a simple interface between the MB86974 MAC and transceivers.

When the MB86974 is transmitting information, data is taken from the host and translated to a 4-bit data nibble stream and transferred through the MII. When the transceiver is receiving information, the incoming 4-bit nibble stream is presented through the MII to the MB86974.

MII Station Manager (MSM)

The MII Station Manager has a serial management interface that allows access to control and status MII registers in the transceiver. This interface consists of: a pair of signals that physically transport the management information frames across the MII; and a register set that can be read and written using these frames.

The MB86974 100/10-Mbit/s Ethernet MAC processes the station Management Data (MDIO and MDC) signals of the MII controller, but does not interpret the values. The MII station manager provides logic for reading and writing control and status registers in a configured Physical layer (PHY) device through the MII-defined serial interface.

If a specialized application–such as a bridge, router, or switched hub–requires access to these values to negotiate configurations, these values are available via MAC control and status registers that trigger read and write operations across the station management data interface. For more information, see page 29.

7-Wire Transceiver Interface

The 7-wire interface allows for a connection to older 10Mb/s transceivers. A simple interface circuit using the MB86961A converts these signals to the standard Attachment Unit Interface (AUI), for example, to connect to 10BASE2 or 10BASE5 networks. An example may be found on page 62

EEPROM Controller

The EEPROM controller provides logic for reading and writing an optional external EEPROM or ROM device. The external devices supported are the standard 4-wire devices such as the MicroChip 93LC46B and the National NM93C46. The controller also supports timing-compatible devices, smaller devices, and read-only equivalent devices. For more information, see the table on page 30.

Fields of an IEEE 802.3/Ethernet Packet (Frame)

Packet (encoded on the medium)							
Added by transmitter, stripped by receiver		Data Frame (sent by user)				Added by transmitter	
		Data Frame (delivered to user)				Optionally stripped by receiver	
Preamble	SFD	Destination Address	Source Address	Length or Type	LLC Data	Pad Bytes	CRC
7 bytes	1	6 bytes	6 bytes	2	0-1500	0-46	4 bytes
				high low	...	...	msb lsb

MAC Frame and Packet Formats

The table above shows the fields of an IEEE 802.3/Ethernet packet. The standard packet has the following fields:

- *Preamble*–7 identical bytes. The bits in each byte are 10101010, transmitted from left to right.
- *Start Frame Delimiter* (SFD)–1 byte. The bits are 10101011, transmitted from left to right.
- *Destination Address*–6 bytes. This can be an individual or a multicast (including broadcast) address.
- *Source Address*–6 bytes. MAC does not interpret the bytes.
- *Length or Type Field*–2 bytes. Values greater than 1535 are type fields, and values less than 1500 are length fields. For Virtual Local Area Network (VLAN) tagged frames, the length values are increased. The special value 0x8808 is recognized by the MAC as the MAC control frame type.
- *Logical Link Control (LLC) Data*–0 to 1500 bytes.
- *Pad Bytes*–0 to 46 bytes. If the LLC data is less than 46 bytes long, the MAC transmits pad bytes of zeroes. Pad bytes are counted in the length field as part of the LLC data.
- *Cyclic Redundancy Check (CRC)*–4 bytes. A value computed as a function of all fields except the preamble, the SFD, and the CRC itself. The IEEE 802.3 standard also refers to the CRC as the Frame Check Sequence (FCS).

The MAC transmits the least significant bit of each byte first for all fields except the CRC. Throughout this data sheet, the word *packet* denotes all of the bytes transmitted and received; whereas, *frame* refers to the bytes delivered by the user for transmission and to the user who is receiving.

Several factors and options can affect this standard IEEE 802.3 packet:

- Some PHYs can deliver a longer or shorter preamble. The preamble may shrink due to repeaters. The MB86974 MAC transmits the standard format, and can accept zero or more preamble bytes, followed by SFD
- *Short packet* mode allows LLC data fields with less than 46 bytes. There are options to suppress padding on transmit and to allow reception of short packets.
- *Long packet* mode allows LLC data fields with more than 1500 bytes. There is an option to allow reception of long packets.
- *No CRC* mode suppresses the appending of a CRC field. This can be useful in bridging applications, where end-to-end CRC checking is desired.
- *Ignore CRC* mode allows the reception of packets without valid CRC fields. This is useful in network diagnosis, and in some bridging applications.

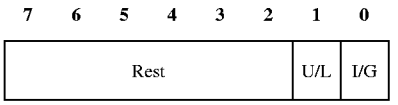
Destination Address Format

Bit 0 of the first byte of the destination address is an address type designation bit. If set to 0, it identifies the address as an individual address; if set to 1, a group address. Group addresses are also called multicast addresses. Individual addresses are also known as unicast addresses. The broadcast address is a special group address, namely FF-FF-FF-FF-FF-FF in hex. There is a special broadcast address used in connection with the full duplex PAUSE operation: 01-80-C2-01-00-01.

Bit 1 of the same byte distinguishes between locally or globally administered addresses. For globally administered (or U, universal) addresses, the bit is set to 0. If an address is to be assigned locally, this bit is set to 1. For the broadcast address, this bit is also a 1.

Note that source addresses are always individual addresses.

Destination address, first byte:



I/G	Individual or Group Flag	=0 Individual address. =1 Group address.
U/L	Universal or Local Flag	=0 Universal address. =1 Local address.
Rest	Rest of Byte	Rest of first byte of destination address.

MB86974 REGISTER SET

The following tables describe the user accessible registers for the Flow Control 100/10-Mbit/s Ethernet controller. Registers are grouped by function:

- PCI configuration registers
- DMA control and status registers
- Flow control registers
- MAC control and status registers

In normal operation, most registers do not need to be accessed directly. Transmit and receive operations use continuous cyclic queues, or rings. Control and status information is communicated through data structures associated with the data itself, as described in "Memory Organization" on page 31. DMA control registers need to be accessed at initialization. Before starting operation, the MAC registers need to be accessed for special configuration needs, including setting the CAM to do address filtering. For an interrupt-based driver, some of the DMA and MAC registers are accessed in the interrupt handler. This allows the driver to enable and disable interrupts, to determine the cause of an interrupt, and to clear interrupt condition bits. The driver can access flow control registers to monitor the progress of local and remote PAUSE commands.

The 'Default State' listed for a register is the value after reset.

There are three basic types of register access:

RO: Read Only. Writing to a RO register has no effect.

R/W: Read and Write. The user can read and write the register. The controller may also write the register, and some parts may be reserved or read only, so the user may not always be able to read what was written.

RClr/W: Read with Clear and Write. Reading the register clears the value. It is used for counters and can also be written to change the count period.

Registers with access marked with "*" have special semantics, such as bits that are "write 1 to clear," and so forth, as explained in the detailed descriptions.

Reserved bits are initialized to 0. To ensure that these bits are compatible with possible future uses, software should leave them unchanged when writing to registers. Software should not depend on the value of reserved fields being 0.

MB86974 PCI Configuration Register Map (in PCI Configuration Address Space)

Address	Register Name	Bytes	Access	Default State	Note
00h	Vendor ID	2	RO	10CFh	Constant
02h	Device ID	2	RO	2005h	Constant
04h	PCI Command	2	R/W	0000h	HWReset
06h	PCI Status	2	R/WC	0200h	SWReset
08h	PCI Class Code	4	RO	0200_0001h	Constant
0Ch	PCI Control	4	R/W	8000_0000h	HWReset
10h	I/O Base Address	4	R/W	0000_0001h	HWReset
14h	Memory Base Address	4	R/W	0000_0000h	HWReset
2Ch	Subsystem Vendor ID	2	R/W	Adapter/Oh	EEPROM
2Eh	Subsystem ID	2	R/W	Adapter/Oh	EEPROM
3Ch	PCI Interrupt	4	R/W*	0000_0100h	

PCI Configuration Registers

The number of configuration address spaces reserved for each PCI device is 256. The first 64 address spaces are predefined by the PCI specification. The Flow Control 100/10-Mbit/s Ethernet controller uses only the first 64 address spaces. These are all addressed via the IDSEL pin. (The other register groups are all acessed via Memory or I/O space as set by the PCI Command Register).

Vendor ID Register (00h)

The Vendor ID Register identifies the component manufacturer. It contains a unique identification number assigned by PCI SIG. The value for the MB86974 is Fujitsu's number 10CFh.

Device ID Register (02h)

The Device ID Register identifies the specific device. The 16-bit number is assigned by Fujitsu and is 2005h for the MB86974. For revisions, see the Class Code Register on page 14.

PCI Command Register (04h)

The PCI Command Register describes how the controller can generate or respond to PCI cycles (see the PCI Specification, Section 6.2.2, "Device Control"). For the controller to operate correctly, Bus Master and either Memory Space or I/O Space must be set. System Error Enable and Parity Error Response are optional. The controller ignores Fast Back to Back Enable, Wait Cycle Control, VGA_Palette Snoop, Memory Write Invalidate Enable, and Special Cycles, which are always 0. The controller only contains one address comparator that is shared between memory and I/O accesses as follows: when MemS=0 and IOS=1, the controller responds to I/O space accesses; when MemS=0 and IOS=0, the controller does not respond to any accesses; when MemS=1, the controller responds to memory space accesses only regardless of the value of IOS. When MemS and IOS are initialized, the corresponding Base Address Register (10h or 14h) must also be initialized.

Hardware reset to value 0000h. Software reset has no effect.

PCI Command Register Bit Assignments (04h)

Bit(s)	Name	Description	Default State	Access
15-10	Reserved	Write to 0, ignore read.	00	
9	Fast Back-to-Back Enable	=0. Controller cannot do fast back-to-back transactions to different devices.	0	RO Const
8	System Error Enable	Enable the system error ($\overline{SERR}$) driver.	0	R/W
7	Wait Cycle Control	=0. Controller does not generate address/data stepping.	0	RO Const
6	Parity Error Response	Device is to respond to parity errors.	0	R/W
5	VGA Palette Snoop	=0. No special VGA palette snooping.	0	RO Const
4	Memory Write and Invalidate Enable	=0. Controller does not generate Memory Write and Invalidate commands.	0	RO Const
3	Special Cycles	=0. Controller ignores special cycle operations.	0	RO Const
2	Bus Master	Device can behave as a bus master.	0	R/W
1	Memory Space	Device can respond to memory accesses.	0	R/W
0	I/O Space	Device can respond to I/O space accesses.	0	R/W

PCI Status Register Bit Assignments (06h)

Bit(s)	Name	Description	Default State	Access
15	Detected Parity Error	Parity error was detected. Set even if the Parity Error Response bit in the PCI Command Register is not set. Parity errors during address cycles are included.	0	R/WC
14	Signaled System Error	Set when the device asserts the $\overline{SERR}$ line.	0	R/WC
13	Received Master Abort	This device was master, and a master transaction was terminated with master-abort, except for a special cycle.	0	R/WC
12	Received Target Abort	This device was master, and a master transaction was terminated with target-abort.	0	R/WC
11	Signaled Target Abort	This device was target, and a master transaction was terminated by asserting target-abort.	0	R/WC
10:9	Device Selection Timing	=01. The controller's slowest $\overline{DEVSEL}$ timing as a target device is "medium." The bits are encoded: 00 - Fast, 01 - Medium, 10 - Slow	01b	RO Const
8	Data Parity Detected	Set when (1) $\overline{PERR}$ asserted or observed by the controller, (2) agent setting $\overline{PERR}$ acted as master, and (3) Parity Error Response bit in PCI Control Register is set.	0	R/WC
7	Fast Back-to-Back Capable	=0. The controller is not capable of accepting fast back-to-back transactions from different agents.	0	RO Const
6:0	Reserved	Write to 0, ignore read.	0	R/W

The PCI Status Register records status information for PCI-bus-related events. It has some special operating characteristics. For example, individual bits are cleared by writing a 1; writing a 0 has no effect. This is necessary to support simultaneous updating of status information by both the system and the PCI device. For a more complete description, see the PCI Specification, Section 6.2.3, "Device Status."

Software reset to value 0200h.

PCI Class Code Register Bit Assignments (08h)

Bit(s)	Name	Description	Default State	Access
31-24	Base Class ID	Set to 02h for Network controller.	02h	RO
23-16	Sub Class ID	Set to 00h for Ethernet controller.	00	RO
15-8	Programming Interface	Set to 00h. No register level programming interfaces defined.	00	RO
7-0	Revision ID	Set by Fujitsu to 01h, current revision number, an extension to the Device ID register.	01h	RO

PCI Control Register Bit Assignments (0Ch)

Bit(s)	Name	Description	Default State	Access
31-24	Built-in Self Test	Controls invocation of self test code on startup.	80	R/W*
23-16	Header Type	Set to 00h for single function, standard layout of bytes 10h through 3Fh in configuration space.	00	RO
15-8	Latency Timer	Number of PCI bus clocks for controller as bus master.	00	RO
7-0	Cache Line Size	System cache line size.	00	R/W

The Built-in Self Test (BIST) is used to test the on-chip buffers. The high-order bit of BIST (bit 31 of PCI_CTL) is a R/O 1, indicating that the controller supports BIST. The next bit (bit 30 of PCI_CTL) is used to start the test. Writing a 1 to bit 30 invokes the test; bit 30 is cleared when the test is complete.

If the test fails, the low order bits of the BIST status (bits 25, 24 in PCI_Ctl) are set to indicate the nature of the error: bit 25 indicates a RAM parity error; bit 24 indicates a RAM test failure; that is, read data did not match expected data. Bits 29 to 26 are reserved. Software drivers should invoke BIST during initialization. Invoking BIST overwrites the RAM based registers. With a 33 MHz clock, the BIST test takes approximately 123us, for the 1K double-word memory.

The default value of the Latency Timer is 0, indicating that it is programmable.

On hardware reset, Cache Line Size is set to zero. Software drivers are responsible for setting an appropriate default Cache Line Size value. Cache line sizes are used to select Memory Read Multiple commands or Memory Read Line commands for burst reads. The recommended size for most systems is eight double words (32 bytes). The controller can support sizes up to 127 double words.

Hardware reset to value 8000_0000h. Software reset has no effect.

I/O and Memory Base Address Registers (10h & 14h)

These registers are used to map the DMA and MAC control and status registers into either I/O address space or system memory space. Both I/O address space and memory address space are limited to 32 bits. When setting a Base Address Register, the corresponding control bit in the PCI Command Register must also be set.

For each register, bits 31:4 contain the upper 28 bits of the base address. Bit 0 is read-only. It contains a 1 for the I/O space address and 0 for the memory space address. Bits 3:1 of the memory base address are read-only 00h, to indicate that the memory base address must be a multiple of 16. It may be anywhere in the 32-bit address space.

A hardware reset sets the base address values to 0000_000xh.

Subsystem Vendor and ID Registers (2Ch & 2Eh)

After a hardware reset, the Subsystem Vendor ID and Subsystem ID registers are initialized with values read from the EEPROM/ ROM, if present. An attempt to read these registers while they are being loaded will be terminated with a target retry.

The Subsystem Vendor ID Register identifies the adapter card manufacturer. It contains a unique identification number assigned by PCI SIG. The default value is 0. If an EEPROM is present, it is used to set the value after a hardware reset.

The Subsystem ID Register identifies the specific device. The adapter card manufacturer assigns the 16-bit number. Vendors that produce both devices and subsystems must ensure that subsystem ID values are different from device ID values. The default value is 0. If an EEPROM is present, it is used to set the value after a hardware reset.

PCI Interrupt Register Bit Assignments (3Ch)

Bit(s)	Name	Description	Default State	Access
31-24	Maximum Latency	=0 (unit = 1/4 microsecond) Desired setting for latency timer values.	00	RO
23-16	Minimum Grant	= 0 (unit = 1/4 microsecond) Minimum burst period, assuming 33 MHz clock.	00	RO
15-8	Interrupt Pin	Set to 01h for $\overline{\text{INTA}}$.	01	RO
7-0	Interrupt Line	Set by system. Interrupt line routing information.	00	R/W

The value of 0 for Maximum_Latency and Minimum_Grant indicates that these registers are not used for determining these values.

To determine if an interrupt is being generated by this PC device, software should examine the Interrupt Source Register. For more information, see "Interrupt Source Register (24h)" on page 20.

DMA Control and Status Register Map

Address	Register Name	Bytes	Access	Default	Note
00h	DMA Control	3	R/W	0000_1020h	HWReset
04h	Transmit Frame Pointer	4	R/W	0000_0001h	SWReset
08h	Transmit Threshold	2	R/W	-	no Reset
0Ch	Transmit Polling Counter	2	R/W	-	no Reset
10h	Buffer List Frame Pointer	4	R/W	0000_0001h	SWReset
14h	Receive Fragment Size	2	R/W	0000_0000h	HWReset
18h	Interrupt Enable	4	R/W	0000_0000h	HWReset
1Ch	Free Descriptor Area Base	4	R/W	0000_0000h	HWReset
20h	Free Descriptor Area Limit	2	R/W	0000_0000h	HWReset
24h	Interrupt Source	2	R/W*	0000_0000h	SWReset

The DMA engine and system software jointly manage three queues, whose pointer and control registers are in this group. These are the Transmit Queue - TxQ, the Receive Queue - RxQ, and the Buffer List -BL_Q.

The *transmit queue* is a list of frame descriptors that are ready for transmission. The *receive queue* is a list of frame descriptors that have been received and are ready for processing by the system software. The *buffer list* is a list of buffer descriptors that describe areas of system memory in which to store received data.

The Free Descriptor Area (FDA) is the memory area where the controller writes the frame descriptors and buffer descriptors for the receive queue.

DMA Control Register

The DMA Control Register controls the transfer of data in master mode: burst size; big-endian byte ordering; and test mode functionality. In addition, the register controls power management wake-up and software-interrupt functions. The controller provides software interrupt as a service for software drivers.

The Transmit Wake Up bit supports immediate data transmission instead of waiting for the current polling cycle to end. If set, and if the transmitter is polling, the current polling cycle is terminated. This bit is cleared at the end of the current polling cycle.

The Transmit and Receive Big Endian bits support the transmission and reception of data that has been ordered for a big endian machine. Note that only data (bytes in the areas pointed to by the buffer descriptors) are affected. Control information, which includes registers, frame descriptors, and buffer descriptors, are always in native PCI bus, or little endian format.

DMA Control Register Bit Assignments (00h)

Bit(s)	Name	Description	Default State	Access
31-19	Reserved	Write to 0, ignore read.	0000	R/W
18	Interrupt Mask	When set, causes interrupt signals to be disabled.	0	R/W
17	Software Interrupt Request	When set, cause an interrupt to be signaled.	0	R/W
16	Transmit Wake Up	When set, abort the current polling cycle and begin transmission.	0	R/W
15	Receive Big Endian	If set, treat receive data as big endian.	0	R/W
14	Transmit Big Endian	If set, treat all transmit data as big endian.	0	R/W
13	Test Mode	Enable test mode functions.	0	R/W
12	Power Management	Enable dynamic power management.	1	R/W
11:9	Reserved	Write to 0, ignore read.	0	R/W
8:0	DMA Burst Size	Size of data bursts requested in master mode.	020	R/W*

Test Mode enables certain test features, such as the ability to read and write all of internal DMA RAM. See "CAM Address Register (60h)" on page 29 for details. The Power Management bit controls dynamic power management. The default value is 1, or enabled.

The DMA Burst Size field controls the size of data transfers requested across the PCI Bus when in master mode. It is a nine-bit register, with the two low-order bits forced to zero, i.e. values must be a multiple of four. The default value after hardware reset is 32 bytes or 8 double words. This can be modified by the software drivers. It cannot be set to zero; an attempt to write a zero is ignored. Generally, the DMA Burst Size register should be a multiple of the PCI cache line size value (see page 15). Care must be used with burst sizes of 4, 8, and 12 in 100-Mb/s full duplex mode.

Hardware reset to the value 0000_1020h. Software reset has no effect.

DMA Transmit Frame Pointer (04h)

The Transmit Frame Pointer Register contains the address of the first frame descriptor to transmit. After software reset, the EOL bit is set. To enable transmission or polling for packets to transmit, the system must set this register to a properly initialized frame descriptor. A valid address must be aligned to a 16-byte boundary; that is, bits 0 to 3 must be zero.

For a description of how polling is controlled, see "DMA Transmit Threshold Control Bit Assignments (08h) DMA Transmit Polling Control (0Ch)" on page 18.

Software reset to value 0000_0001h.

Bit(s)	Name	Description	Default State	Access
31-4	Transmit-Frame Pointer	Pointer to first frame descriptor to transmit.	0000_000h	R/W
3:1	Reserved	Write to 0	0h	(R/W)
0	EOL	End of List flag.	1	R/W

DMA Transmit Threshold Control (08h)

The Transmit Threshold Control Register controls the buffer latency for transmitting packets. If the threshold value is non-zero, then data transfer to the MAC begins as soon as the DMA transmit FIFO contains this number of bytes or when a complete packet is in the FIFO. If the threshold value is zero, data transfer to the MAC starts immediately. Software drivers must initialize this register, optionally by software transfer from the EEPROM/ROM.

If the threshold value is set too low, the DMA transmit FIFO can run dry due to PCI bus latency. If this occurs, as indicated by the MAC transmit status, system software should increase the TxThreshold value. Do not set the threshold value greater than 1700. In long-packet-mode, this can cause buffer memory to fill without enabling transmission thus causing the transmitter to hang.

Held in internal RAM, hardware and software reset have no effect.

DMA Transmit Threshold Control Bit Assignments (08h)

Bit(s)	Name	Description	Default State	Access
31-11	Reserved	Write to 0, ignore read.	00	R/W
10:0	Tx Threshold	Transmit Threshold value.	00	R/W

DMA Transmit Polling Control (0Ch)

The Transmit Polling Control Register controls how often the controller polls for packets to transmit. If the value is zero, polling is not performed; otherwise, an internal counter is set to this value, which decrements to zero. When the register reaches zero, a read is done to see if a new transmit packet has arrived. With a 33 MHz clock, each unit in the polling counter is equivalent to 61.44 micro-seconds. Software drivers must initialize this register. The value can be specified in the EEPROM/ROM, with software driver support.

It is held in internal RAM, so hardware and software reset have no effect.

Bit(s)	Name	Description	Default State	Access
31-12	Reserved	Write to 0, ignore read.	00	R/W
11:0	Tx Polling Counter	Transmit Polling Cntrl value, in 61.44 usec units.	00	R/W

DMA Buffer List Frame Pointer (10h)

The Buffer List Frame Pointer Register contains the address of the first frame descriptor to read for acquiring free buffer descriptors. The system must set this register to a properly initialized frame descriptor to enable reception. A valid address must be aligned to a 16-byte boundary; that is, bits 0 to 3 must be zero.

Software reset to value 0000_0001h.

Bit(s)	Name	Description	Default State	Access
31-4	Buffer List Frame Pointer	Pointer to first frame descriptor to transmit.	00	R/W
0	EOL	End of List flag.	01	R/W

DMA Receive Fragment Size Register (14h)

The Receive Fragment Size Register specifies the smallest data fragment that the controller will generate. The size must be a multiple of four; that is, the two low order bits are always zero. Packing can be enabled globally using the Enable Packing bit, or on a per buffer area basis, as explained in “Frame Descriptor Control Field (FDCtl)” on page 33.

The controller always begins storing received data on a 4-byte-aligned address. There may be 1 to 3 unused bytes at the end of a frame, due to alignment.

When packing is enabled, the Min_Frag value must be greater than zero for the controller to work. If packing is desired, software drivers must set the Minimum Fragment field and the EnPack bit.

Hardware reset to 0000_0000h. Software reset has no effect.

Bit(s)	Name	Description	Default State	Access
31-16	Reserved	Write to 0, ignore read.	0000	R/W
15	EnPack	Enable Packing using Min Frag (vs FDCtl).	0	R/W
14:12	Reserved	Write to 0, ignore read.	0	R/W
11:0	Min_Frag	Min number of bytes to write (multiple of 4).	000	R/W

Interrupt Enable Register (18h)

The Interrupt Enable Register controls the generation of interrupts in response to errors and other conditions detected by the DMA engine.

The Early Notify Enable bit supports applications that minimize latency. Note that the Frame Descriptor will not be valid when the Early Notify is processed. Only the first buffer descriptor is valid when this interrupt is signaled.

Hardware reset to 0000_0000h. Software reset has no effect.

Bit(s)	Name	Description	Default State	Access
31-12	Reserved	Write to 0, ignore read.	0000_0	R/W
11	Non-Recoverable Abort Enable	Interrupt when there is an internal non-recoverable abort condition.	0	R/W
10	Transmit Control Complete Enable	Interrupt when transmission of a MAC control packet is complete.	0	R/W
9	DMA Parity Error Enable	Interrupt if a parity error is detected in reading or writing the DMA internal RAM.	0	R/W
8	Data Parity Detected Enable	Interrupt if bit 8 of PCI Stat Register is set.	0	R/W
7	Early Notify Enable	Interrupt after writing the first buffer and buffer descriptor of a packet.	0	R/W
6	Detected Parity Error Enable	Interrupt if the controller detects a parity error on a PCI bus data transfer during a master access.	0	R/W
5	Signaled System Error Enable	Enable an interrupt if the controller signals system error.	0	R/W
4	Received Master Abort Enable	Interrupt if the controller receives a master abort while bus master.	0	R/W
3	Received Target Abort Enable	Interrupt if the controller receives a target abort while acting as bus master.	0	R/W
2	Signaled Target Abort Enable	Interrupt if the controller signals a target abort while target.	0	R/W
1	Buffer List Exhausted Enable	Interrupt if the Buffer List becomes exhausted; if the controller encounters a descriptor in the BL still owned by the system.	0	R/W
0	Free Descriptor Area Exhausted Enable	Interrupt if Free Descriptor Area becomes exhausted; if the controller encounters a block in the FDA still owned by the system.	0	R/W

Free Descriptor Area Base Register (1Ch)

The Free Descriptor Area Base Register contains the starting address of the area reserved for the controller to write frame and buffer descriptors for received packets. The address must be a multiple of 16 bytes; that is, bits 0, 1, 2, and 3 are zero.

Hardware reset to value 0000_0000h. Software reset has no effect.

Free Descriptor Area Limit Register (20h)

The Free Descriptor Area Limit Register contains the count of the number of 16-byte blocks in the receive descriptor area, in bits 15:4. Alternatively, the low 16 bits can be viewed as a byte offset from the base, which must be a multiple of 16.

Each 16-byte block holds one frame descriptor, or two 8-byte buffer descriptors, so the maximum size of a single descriptor area is 4095*16 or 64K - 16 bytes.

The FDA Limit Register must point to the lowest offset in the FDA where a new frame descriptor can be safely begun. Enough space must be allowed for a maximum-size packet to have enough space for a maximum number of buffer descriptors. See "Queue Initialization" on page 42 for more details.

Hardware reset to value 0000_0000h. Software reset has no effect.

Interrupt Source Register (24h)

The Interrupt Source Register is read by system software to see if there is an interrupt associated with the Ethernet controller. The register also provides status bits for some conditions that are not reported elsewhere. If bits 10 through 0 are all zeroes, the controller did not generate an interrupt. If an interrupt is associated with the Ethernet controller, all further interrupts from the Ethernet controller can be masked with the IntMask bit of the DMA Control Register.

Bit 8 is set if a single Frame Descriptor Register requires more than 28 buffer descriptors. Bit 13 is set whenever a DMA RAM parity error is detected. Bit 7 is set and an interrupt is generated only if the Detected Parity Error Enable bit of the Interrupt Enable Register is set.

W1Clr: indicates a bit that is cleared by writing 1 to the bit. Writing 0 has no effect.

RO: indicates a read-only bit, which is cleared by clearing the condition that sets the bit or by resetting the controller.

Software reset to value 0000_0000h.

Bit(s)	Name	Description	Default State	Access
31-15	Reserved	Write to 0, ignore read.	0000_0	R/W
14	Non-recoverable Abort	Set if non-recoverable abort occurs.	0	W1Clr
13	DmParErrStat	Set if a DMA parity error occurs.	0	W1Clr
12	BL_Ex	Set if the buffer list is exhausted during reception.	0	W1Clr
11	FDAEx	Set if the free descriptor area is exhausted during reception.	0	W1Clr
10	IntNRAbt	Interrupt caused by a non-recoverable abort state.	0	RO
9	IntTxCtlCmp	IntTxCtlCmp.	0	W1Clr
8	IntExBD	Interrupt caused by excessive (more than 28) buffer descriptors.	0	W1Clr
7	DmParEr	Interrupt caused by a DMA parity error.	0	RO
6	IntEarNot	Interrupt caused by Early Notify.	0	W1Clr
5	SWInt	Interrupt caused by Software Interrupt Request.	0	RO
4	IntBL_Ex	Interrupt caused by Buffer List Exhausted.	0	RO
3	IntFDAEx	Interrupt caused by Free Descriptor Area Exhausted.	0	RO
2	IntPCI	Interrupt reported in PCI Controller, PCI_Stat.	0	RO
1	IntMacRx	Interrupt caused by MAC Receive Status, written to memory.	0	W1Clr
0	IntMacTx	Interrupt caused by MAC Transmit Status, written to memory.	0	W1Clr

Flow Control Register Map

The following table lists the three registers that manage the flow control operation:

Address	Register Name	Bytes	Access	Default	Note
30h	Pause Count	2	RO	0000_0000h	SWReset
34h	Remote Pause Count	2	RO	0000_0000h	SWReset
38h	Transmit Control Frame Status	2	R/W		Internal RAM

Pause Count Register (30h) and Remote Pause Count Register (34h)

The Pause Count Register holds the 16-bit count of time slots that the transmitter is still being paused after a MAC control PAUSE operation packet is received. A value of 0 indicates the MAC is not paused. The two high-order bytes are reserved.

The Remote Pause Count Register holds the 16-bit count of time slots that the remote MAC is still being paused after a PAUSE operation packet is sent. This provides an approximate current value of the remote pause counter, based on when a PAUSE command was sent. The two high order bytes are reserved.

Software resets both registers to value 0000_0000h.

Transmit Control Frame Status Register (38h)

The Transmit Control Frame Status Register provides the status of sending a MAC control packet to a remote station via the SdPause bit of the Transmit Control Register. When the transmission is complete, the software drivers can pick up the status from this register. Software can reset this register by clearing it before initiating the transfer of a MAC control frame. The TxCtlCmpEn bit of the Interrupt Enable Register provides an option to generate an interrupt when MAC control-packet transmission is complete.

The format of this register is the same as that of the Transmit Status Register (4Ch) described on page 26.

It is held in internal RAM, so hardware and software reset have no effect.

MAC Control and Status Register Map

The following table lists the functional group, register name, symbol, address, size, and access type for each MAC control and status register:

Address	Register Name	Bytes	Access	Default State	Note
40h	MAC Control	2	R/W*	0000h	HWReset*
44h	CAM Control	1	R/W	0000h	HWReset
48h	Transmit Control & Status	2	R/W	0000h	HWReset*
4Ch	Transmit Status	3	R/O	00_0000h	SW/Packet
50h	Receive Control	2	R/W	0000h	HWReset*
54h	Receive Status	2	R/O	0000h	SW/Packet
58h	Station Management Data	2	R/W	0000h	SWReset
5Ch	Station Management Control and Address	2	R/W*	0000h	SWReset
60h	CAM Address	2	R/W	0000h	SWReset
64h	CAM Data	4	R/W	CAM Data	use RMW*
68h	CAM Enable	3	R/W	00_0000h	HWReset
6Ch	PROM Control	2	R/W*	0000h	SWReset
70h	PROM Data	2	R/W	0000h	SWReset
7Ch	Missed Error Count	2	RClr/W	0000_0000h	HWReset

These registers are generally loaded by the driver software from an EEPROM or ROM on chip power up or chip reset. Some of these registers, such as the CAM Control and Missed Error Count registers, are accessed by system software drivers while the MAC is active. The MAC Transmit and Receive Control and Status registers are generally controlled by the DMA engine after they are setup.

The MAC layer control registers include a master MAC control register, control and status registers for transmit and receive, control registers for the CAM and the EEPROM, and some error counters for network management.

MAC Control Register (40h)

The Missed Roll and Link Status 10 Mb/s bits are read-only status bits. All the others are control bits. The Missed Roll bit is set when the counter rolls over and is cleared when software reads the Missed Count Register, as described in "Missed Error Count Register (7Ch)" on page 30.

Some PHYs may not support full-duplex operation. MAC loopback overrides the full-duplex bit. Some 10-Mb/s PHYS may use Loop10 to control a different function and use the Link10 signal to indicate a different status condition. The Applications section shows another use for the Loop_10 pin.

In automatic connect mode, activity in the form of receive clock and carrier sense on the 10 Mb/s interface will select the 10 Mb/s endec; otherwise, the MII is selected. For some full-duplex operating environments, the 10 Mb/s 7-wire interface may require software configuration via the Connection Mode bits

Software reset is invoked by setting bit 2, Reset. Bit 2 is cleared after software reset is complete. Software reset does not affect other bits.

Software reset is delayed for three clock cycles. This allows completion of the operation that writes the Reset bit. Software can use the Tx_ctl and Rx_ctl registers to request halt after current network transactions are complete, and then check the relevant status register bits before using reset.

The MAC Control Register is hardware reset to the value 0000h. .

Bit(s)	Name	Description	Default State	Access
31-16	Reserved	Write 0, ignore on Read.	00	R/W
15	Link Status 10 Mb/s	Buffered signal on the Link 10 pin.	0	RO
14	Reserved	Write 0, ignore on Read.	0	R/W
13	Enable Missed Roll	Interrupt when missed error counter rolls over.	0	R/W
12:11	Reserved	Write 0, ignore on Read.	00	R/W
10	Missed Roll	Missed error counter rolled over.	0	RO
9:8	Reserved	Write 0, ignore on Read.	00	R/W
7	Loop 10 Mb/s	Value on the LOOP_10 external signal to the 10-Mbit/s endec.	0	R/W
6:5	Connection Mode	Select the connection mode: 00 = Automatic, (default) 01 = Force 10-Mbit/s endec 10 = Force MII (rate determined by MII clock)	00	R/W
4	MAC Loopback	Cause transmission signals to be presented as input to the receive circuit without leaving the controller.	0	R/W
3	Full Duplex	Allow transmission to begin while reception is occurring.	0	R/W
2	Software Reset	Reset all Ethernet controller state machines and FIFOs.	0	R/W
1	Halt Immediate	Stop transmission and reception immediately.	0	R/W
0	Halt Request	Stop transmission and reception after completion of any current packets.	0	R/W

CAM Control Register (44h)

Bit(s)	Name	Description	Default State	Access
15:5	Reserved	Write 0, ignore on Read.	000	R/W
4	Compare Enable	Enable compare mode.	0	R/W
3	Negative CAM	0 = Accept packets CAM recognizes, reject others. 1 = Reject packets CAM recognizes, accept others.	0	R/W
2	Broadcast Accept	Accept any packet with a broadcast address.	0	R/W
1	Group Accept	Accept any packet with a multicast-group address.	0	R/W
0	Station Accept	Accept any packet with a "unicast" station address.	0	R/W

When CAM compare mode is enabled, the CAM memory is read for destination addresses with which to filter incoming messages. The CAM memory is organized as 21 entries of 6 bytes each, as described in "CAM Address Register (60h)" on page 29, which can be individually enabled and disabled through the "CAM Enable Register (68h)" on page 30.

The filtering is organized by the three types of destination address:

- *Station*. This has an even first byte; for example, 00-00-23-AC-04-FE.
- *Multicast-group*. This has an odd first byte, which is not FF-FF-FF-FF-FF-FF; for example, 01-00-00-23-AC-00.
- *Broadcast*. This is defined to be FF-FF-FF-FF-FF-FF.

For 'normal' operation, the CAM memory will contain those addresses that the MAC should accept. This will include the MAC station address(es), and any address for which the MAC acts as a bridge or forwarding agent, and any multicast group addresses. All these addresses would be enabled, and the 'Compare enable' bit would be set. Any valid packet with one of these destination addresses will then be accepted.

The three Accept bits override CAM rejections. Thus it would be 'normal' to accept any packet with the 'broadcast' address.

To place the MAC in promiscuous mode, accept packets with all three types of destination address by setting all three 'Accept' bits. Alternatively, set the Negative CAM bit but clear the Compare Enable bit. The CAM will fail to recognize all packets and the MAC will accept them.

To reject all packets, clear all bits in CAM_ctl Register.

Hardware reset to value 0000h (that is, reject all packets). Software reset has no effect.

Transmit Control and Status Register (48h)

Bit(s)	Name	Description	Default State	Access
15	Reserved	Write 0, ignore on Read.	0	R/W
14	Enable Completion	Interrupt when the MAC transmits or discards one packet.	0	R/W
13	Enable Transmit Parity	Interrupt if the MAC transmit FIFO has a parity error.	0	R/W
12	Enable Late Collision	Interrupt if a collision occurs after 512 bit times (64 byte times).	0	R/W
11	Enable Excessive Collision	Interrupt if 16 collisions occur in the same packet.	0	R/W
10	Enable Lost Carrier	Interrupt if carrier sense is not detected or is dropped during the transmission of a packet.	0	R/W
9	Enable Excessive Deferral	Interrupt if the MAC defers for MAX_DEFERRAL time: = 0.24288 ms for 100 Mbits/s. = 2.4288 ms for 10 Mbits/s.	0	R/W
8	Enable Underrun	Interrupt if the MAC transmit FIFO becomes empty during transmission.	0	R/W
7	MII 10 Mb Mode	Set by software to enable SQE checking in MII 10 Mb mode.	0	R/W
6	Send Pause	Send a PAUSE command, or other MAC Control frame.	0	R/W
5	No Excessive Defer	Suppress the checking of Excessive Deferral.	0	R/W
4	Fast Backoff	Use faster backoff timers for testing (not for normal operation).	0	R/W
3	Suppress CRC	Do not add the CRC at the end of a packet.	0	R/W
2	Suppress Padding	Do not generate pad bytes for packets with less than 64 bytes.	0	R/W
1	Transmit Halt Request	Halt transmission after completing any current packet.	0	R/W
0	Transmit Enable	If zero, stop transmission immediately.	0	R/W

When the transmission of the MAC control packet is complete, the Send Pause bit is automatically cleared. Writing 0 to this bit has no effect. All other bits exhibit standard read/write operation.

Deferral refers to the half-duplex operation of waiting till any packet being received has ended before transmitting a packet. This is not the same as a collision, which occurs when a transmit packet is started, and then a packet is received. Note that if (in half-duplex mode) the MAC defers for the MAX_DEFERRAL time, and the 'Enable Excessive Deferral' interrupt bit is set, the deferred packet will be dropped. This action may cause problems in some test suites, where a response packet is expected after a jabber packet exceeding this time terminates.

To receive an interrupt after each packet, set the enable completion and all the MAC error enable bits. Interrupts may also be enabled only for specific conditions.

SQE ('heartbeat') checking is automatic over the 7-wire interface.

Hardware reset to value 0000h. Software reset clears TxEn but does not affect other bits.

Transmit Status Register (4Ch)

Bit(s)	Name	Description	Default State	Access
31:17	Reserved	Write 0, ignore on Read.	0000h	R/W
16	Signal Quality Error	No heart-beat signal observed at end of transmission.	0	R/W
15	Transmission Halted	Transmission was halted by clearing TxEn or setting TxHalt.	0	R/W
14	Completion	MAC transmits or discards one packet.	0	R/W
13	Transmit Parity Error	MAC transmit FIFO has detected a parity error.	0	R/W
12	Late Collision	A collision occurs after 512 bit times (64 byte times).	0	R/W
11	Transmit 10-Mb/s Status	= 1 if packet was transmitted via the 10-Mb/s interface. = 0 if packet was transmitted via MII.	0	R/W
10	Lost Carrier Sense	Carrier sense is not detected or is dropped during the transmission of a packet (from the SFD to the CRC).	0	R/W
9	Excessive Deferral	MAC defers for MAX_DEFERRAL: (two maximum size packets) = 0.24288 ms for 100 Mbits/s = 2.4288 ms for 10 Mbits/s	0	R/W
8	Underrun	MAC transmit FIFO becomes empty during transmission.	0	R/W
7	Interrupt on Transmit	Set if transmission of packet caused an interrupt condition. This includes the EnComp transmission complete condition, if enabled.	0	R/W
6	Transmitter Paused	Set if transmission was paused after completion of the current packet.	0	R/W
5	Transmit Deferred	Set if transmission of packet was deferred and no collisions occurred.	0	R/W
4	Excessive Collision	Set if 16 collisions occur in the same packet. Transmission skipped.	0	R/W
3:0	Transmit Collision Count	Count of the collisions in transmitting a single packet. If 16 collisions occur, TxColl will be zero, and ExColl is set.	0	R/W

The transmission status flags are set whenever the corresponding event occurs. In addition, an interrupt is generated if the corresponding enable bit in the Transmit Control Register is set.

Note that if the MAC defers for the MAX_DEFERRAL time, and the 'Excessive Deferral' bit is set, the deferred packet will be dropped. This action may cause problems in some test suites, where a response packet is expected after a jabber packet exceeding this time terminates.

The low-order 5 bits can be read and masked as a single collision count; that is, when Excessive Collision is 1, Transmit Collision Count is 0. If Transmit Collision Count is non-zero, Excessive Collision is 0.

Software reset to value 00_0000h. It is cleared at the beginning of each packet sent.

Receive Control and Status Register (50h)

Bit(s)	Name	Description	Default State	Access
15	Reserved	Write 0, ignore on Read.	0	R/W
14	Enable Good	Interrupt upon receipt of a packet with no errors.	0	R/W
13	Enable Receive Parity	Interrupt if the MAC receive FIFO has detected a parity error.	0	R/W
12	Reserved	Write 0, ignore on Read.	0	R/W
11	Enable Long Error	Interrupt upon receipt of a frame longer than 1518 bytes,* unless the long enable bit is set.	0	R/W
10	Enable Overflow	Interrupt upon receipt of a packet when the MAC receive FIFO is full.	0	R/W
9	Enable CRC Error	Interrupt upon receipt of a packet whose CRC is invalid or, during its reception, the PHY asserts RX_ER.	0	R/W
8	Enable Alignment	Interrupt upon receipt of a packet whose length in bits is not a multiple of eight, and whose CRC is invalid.	0	R/W
7	Reserved	Write 0, ignore on Read.	0	R/W
6	Ignore CRC Value	Do not check the CRC.	0	R/W
5	Pass Control Packets	Enable passing of received MAC Control packets to system.	0	R/W
4	Strip CRC Value	Check the CRC, but strip it from the message.	0	R/W
3	Short Enable	Allow reception of frames shorter than 64 bytes.*	0	R/W
2	Long Enable	Allow reception of frames longer than 1518 bytes.*	0	R/W
1	Receive Halt Request	Halt reception after completing any current packet.	0	R/W
0	Receive Enable	If zero, stop reception immediately.	0	R/W

* The frame lengths above do not include preamble and SFD. See "MAC Frame and Packet Formats" on page 10 for more details.

To receive an interrupt after each packet, set the Enable Good and all the MAC Error Enable bits. Interrupts may also be enabled only for specific conditions, such as specific error flags being set.

Hardware reset to value 0000h. Software reset clears RxEn but does not affect other bit values.

Receive Status Register (54h)

Bit(s)	Name	Description	Default State	Access
15	Reception Halted	Reception interrupted by user clearing RxEn or setting RxHalt.	0	R/W
14	Good Received	Successfully received a packet with no errors. If EnGood = 1, an interrupt is generated on each packet received successfully.	0	R/W
13	Receive Parity Error	The MAC receive FIFO has detected a parity error.	0	R/W
12	Reserved	Write 0, ignore on Read.	0	R/W
11	Long Error	Received a frame longer than 1518 bytes.* Not set if the Long Enable bit in the receive control register is set.	0	R/W
10	Overflow Error	The MAC receive FIFO was full when it needed to store a received byte.	0	R/W
9	CRC Error	CRC at end of packet did not match computed value, or the PHY asserted RX_ER during packet reception.	0	R/W
8	Alignment Error	Frame length in bits was not a multiple of eight, and the CRC was invalid.	0	R/W
7	Receive 10-Mb/s Status	= 1 if packet was received via the 10-Mb/s interface. = 0 if packet was received via MII.	0	R/W
6	Interrupt on Receive	Set if reception of packet caused an interrupt condition. This includes Good Received, if the EnGood bit is set.	0	R/W
5	Control Received	Set if packet received is a MAC Control frame.	0	R/W
4:0	Reserved	Write 0, ignore on Read.	0	R/W

* The frame lengths above do not include preamble and SFD. See "MAC Frame and Format Packet" on page 10 for more details.

The receive status flags are set whenever the corresponding event occurs. Once set, a flag stays set until another packet arrives. In addition, an interrupt is generated if the corresponding enable bit in the Receive Control Register is set. Software is responsible for separating alignment, CRC, and frame too-long errors, and reporting them correctly as management information.

Control Received is set if the packet type is 8808h and the CAM recognizes the address. Software is responsible for checking the PAUSE operation code.

A MAC receive-parity error sets RxPar, if interrupt is enabled.

Software reset to value 0000h. It is cleared at the beginning of each packet received.

Station Management Data Register (58h)

The data in the Station Management Data Register (16-bit) is sent out to or received from the PHY over the MDIO and MDC pins. The MII section of the IEEE 802.3 standard for 100-BASE-T, 100-Mbit/s Ethernet defines the format of the required station management data registers. See specific PHY data sheets for additional hardware-dependent registers.

Software reset to value 0000h.

Station Management Data Control and Address Register (5Ch)

Bit(s)	Name	Description	Default State	Access
15:13	Reserved	Write 0, ignore on Read.	0	R/W
12	Preamble Suppress	If set, the preamble is not sent to the PHY.	0	R/W
11	Busy bit	Set to begin operation; controller clears when operation completes.	0	R/W
10	Write	Set for write, clear for read.	0	R/W
9:5	PHY Address	Address of PHY device to read or write.	0	R/W
4:0	Address	Address inside the PHY of register to read or write.	0	R/W

Before attempting to access the PHY control registers, software should read the Station Management Data Control Address Register to ensure that the Busy bit is not set. The controller provides support for reading and writing of station management data to the PHY. Setting of options in station management registers does not affect the controller. Some PHYs may not support the option to suppress preambles after the first operation.

Software reset to value 0000h.

CAM Address Register (60h)

Bit(s)	Name	Description	Default State	Access
15:12	Reserved	Write 0, ignore on Read.	0	R/W
11:0	CAM Address	The address of the 4-byte CAM location to read or write.	0	R/W

In normal operation, the CAM Address and CAM Data registers can read or write the CAM contents, including the three double-word locations immediately after the CAM for flow control operation, as shown in "CAM Memory Map" on page 37. Writing to other memory locations in normal operation has no effect.

When the TestMode bit of the DMA Control Register (see page 17) is set, the CAM Address can be used to read or write the entire DMA_RAM.

Software reset to value 0000h.

CAM Data Register (64h)

Bit(s)	Name	Description	Default State	Access
31:24	CAM Data[0]	The data byte to read or write to CAM location.	0	R/W
23:16	CAM Data[1]	The data byte to read or write to CAM location.	0	R/W
15:8	CAM Data[2]	The data byte to read or write to CAM location.	0	R/W
7:0	CAM Data[3]	The data byte to read or write to CAM location.	0	R/W

Four bytes are accessed each time there is a read or write of the CAM Data Register; that is, the PCI Byte Enables are ignored. System software must take care to perform read, modify, and write logic when modifying only 2 bytes of a 4-byte word.

The CAM Data Register has a copy of the data stored at the CAM bytes addressed by the CAM Address Register. The register can be read more than once. When data is written to this register, the addressed CAM bytes are changed.

Unlike data transferred in master mode, data transferred via this CAM interface is always interpreted internally as big-endian.

CAM Enable Register (68h)

Bit(s)	Name	Description	Default State	Access
31:21	Reserved	Write 0, ignore on Read.	0	R/W
20:0	CAM Enable	The list of valid CAM entries.	0	R/W

The CAM Enable Register indicates which entries are valid for address filtering. Up to 21 entries, numbered 0-20, can be active.

Hardware reset to value 00_0000h. Software reset has no effect.

PROM Control Register (6Ch)

Bit(s)	Name	Description	Default State	Access
15	Busy bit	Set to begin operation. Will be cleared by the serial driver when operation is complete.	0	R/W
14:13	Operation Code	1 0 = Read 0 1 = Write 0 0 = Enable or Disable Writing, as specified in PROM_Address: [5:4] = 11, Enable [5:4] = 00, Disable 1 1 = Erase	0	R/W
12:6	Reserved	Write 0, ignore on Read.	0	R/W
5:0	PROM_Address	Allows addressing of up to 64 16-bit entries.	0	R/W

The PROM Control Register provides control and status information and buffering for the PROM controller, which controls the reading and writing of an optional external EEPROM or small serial ROM device.

The EEPROM or ROM is clocked at 1/34 of the PCI clock. For timing, see "EEPROM Timing" on page 59.

Before attempting to access the EEPROM/ROM memory, software should read the PROM Control Register to ensure that the Busy bit is not set.

Software reset to value 0000h.

PROM Data Register (70h)

The PROM Data Register provides the 16 bits of data written to or read from PROM. The EEPROM Controller supports the MicroChip 93LC46B, the National NM93C46, and compatible devices.

Upon hardware reset, the Subsystem Vendor ID and Subsystem ID registers are loaded from the first two locations in the EEPROM or ROM. Software drivers are responsible for reading the station address (usually stored in the EEPROM), storing it in the CAM, and enabling CAM operation as part of driver initialization.

Software reset to value 0000h.

Missed Error Count Register (7Ch)

The 16-bit Missed Error Count Register provides a count of the number of valid packets that are rejected by the MAC unit because the MAC receive FIFO overflows, a parity error occurs, or the Receive Enable bit (Rx_en) is cleared. This count excludes packets the CAM rejects. Together with status information for packets transmitted and received, these counters provide the information needed for station management. When this register rolls over from 0x7FFF to 0x8000, it sets the Missed Roll bit in the MAC Control Register. It also generates an interrupt if the Enable Missed Roll bit is set. It is the responsibility of software to maintain a larger global count. If station management software requires more frequent interrupts, this register can be set to a value closer to the rollover value of 0x7FFF. For example, setting the register to 0x7F00 provides for an interrupt after counting 256 occurrences.

Hardware reset to value 0000_0000h. Cleared when read.

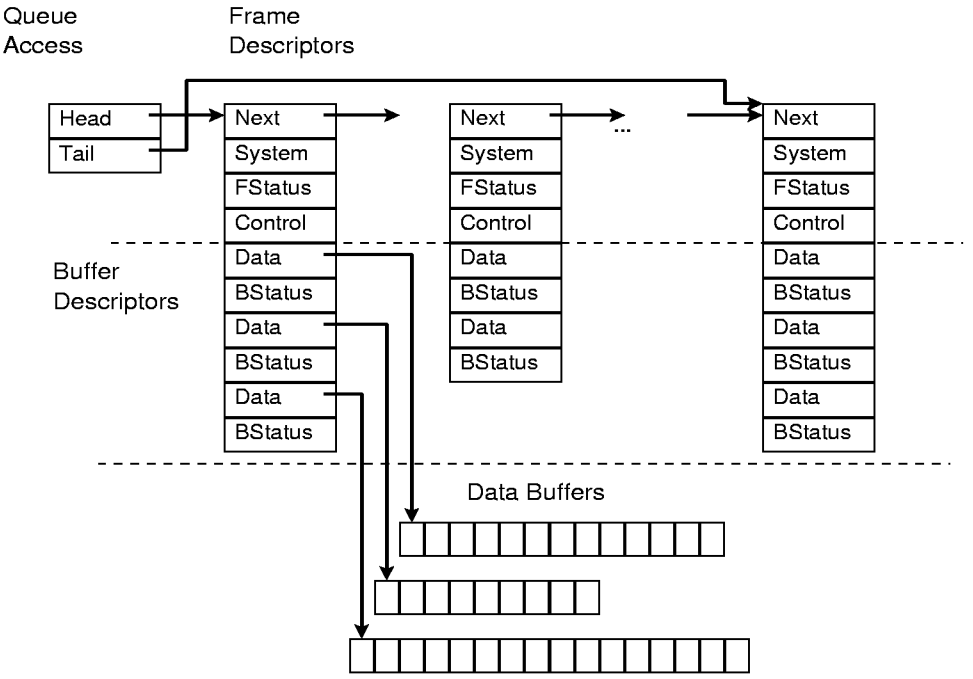
Memory Organization

The data structures used by the MB86974 PCI-based Flow Control 100/10-Mbit/s Ethernet controller to communicate with the host system are located in system memory (on the host).

The three types of basic data structures (frame descriptors, buffer descriptors, and data buffers) are used in three ways:

1. *Transmit queue*: a list of frame descriptors for packets to transmit
2. *Receive queue*: a list of frame descriptors for packets that have been received
3. *Buffer list*: a list of frame descriptors with unused buffers for receiving data

The following diagram shows how each structure is organized. The next sections describe them. Subsections describe how some structures contain different information when used in different queues.



In continuous polling operation, a queue does not become empty once it is active. There is always a "dummy" frame descriptor at the end of the list that belongs to the producer of new descriptors.

For a detailed explanation, see "Queue Initialization" on page 42.

To begin transmission, the system stores into the Transmit Frame Pointer Register the address of the first frame descriptor in the transmit queue. The controller traverses the transmit queue, updating the status of transmitted packets. "Transmission complete" appears in the frame descriptor status field and in the ownership bit of the frame descriptor control field. This allows system software to process the queue after transmission; for example, to free buffers.

The controller acquires buffers from the buffer list and writes new frame descriptors and new buffer descriptors into the free descriptor area, as described in "Queue Initialization" on page 42.

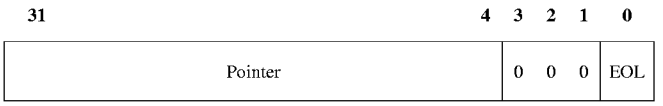
Frame Descriptors

Each frame descriptor has a 32-bit pointer to the next frame descriptor in the queue, a 32-bit system data field, a 16-bit length field for this frame, and 16-bit control and 24-bit status fields. The controller preserves the contents of the frame system data field, FDSystem. Either the system or the application programs can use this field. The initial value for frame descriptors written on the receive queue is obtained from the list frame descriptor of the current buffer list. Each queue makes slightly different use of the FDNext, FDCtl, FDStat, and FDLength fields, as explained in the next sections. The following table shows the layout of a frame descriptor:

Byte 3	Byte 2	Byte 1	Byte 0	Offset
FDNext				00
FDSystem				04
Reserved	FDStat			08
FDCtl		FDLength		0C

Next Frame Descriptor Field (FDNext)

The next frame descriptor field contains either an EOL flag (=1 for end of list), or a pointer to the next frame descriptor in the same queue. Frame descriptors must be aligned to 16-byte boundaries; that is, a valid pointer must have bits 0-3 set to zero. On all the queues, the next frame descriptor field is used to stop the consumer of the list by setting the EOL bit. The consumer must wait for the producer of the list to clear the EOL bit when it stores a valid pointer. On the buffer list queue, chaining from one buffer pool to another using the next frame descriptor field is possible. If chaining of buffer lists is not used, the software drivers should set the FDNext field to contain its own address. This would cause the controller to re-examine the same buffer area for reuse. Alternatively, the EOL bit can be set, causing the controller to stop.



Frame Descriptor System Field (FDSystem)

The FDSystem field is a 32-bit field reserved for system software use. It could be a pointer to a table of information, a pointer to C++ virtual functions, and so forth. On the transmit queue, the FDSystem field is not used. On the receive queue, the controller copies the contents of the FDSystem field from the current buffer list queue, where the first buffer descriptor was allocated.

Frame Descriptor Status Field (FDStat)

On the transmit and receive queues, the FDStat field is used for reporting transmission and reception completion status. For a description of the status bits, see the "Transmit Status Register (4Ch)" on page 26 and "Receive Status Register (54h)" on page 28.

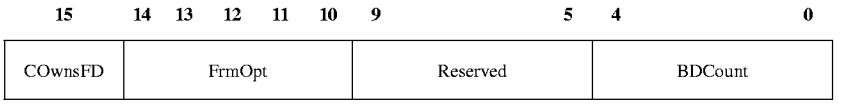
On the receive buffer list, the FDStat field is not used.

Frame Descriptor Length Field (FDLength)

On the transmit queue, the FDLength field is not used. On the receive queue, the controller sets the FDLength field to the total length of the frame. On the buffer list queue, the FDLength field is used to count the number of free buffer descriptors allocated to the queue. The controller accesses the buffer list frame descriptor via the Buffer List Frame Pointer Register. If the controller encounters a buffer it does not own, it sets the BL_Ex bit in the Interrupt Source Register (see page 20) and waits for the system to clear it. The controller reads the buffer descriptors, using the FDLength field as a limit. When the controller nears the end of the list, it gets the next frame descriptor pointed to by the FDNext field as described in "Free Descriptor Area Base Register (1Ch)" on page 19.

Frame Descriptor Control Field (FDCtl)

The following tables show the abbreviation, field name, description, and usage of the FDCtl field:



Symbol	Name	Description	Used by
COwnsFD	Controller Owns Frame Descriptor	=1. The controller owns the frame descriptor, after the system sets COwnsFD. =0. The system owns the frame descriptor, after the controller clears COwnsFD.	Tx, Rx
FrmOpt	Frame Options	Per Frame Control Options. See below.	Tx, BL
BDCount	Buffer Descriptor Count	Number of BD's allocated. (0 to 29).	Tx, Rx

The ownership field is used in the transmit and receive queues to synchronize processing by the controller and the system. Frame options are used in the transmit queue and the buffer list. The buffer descriptor count field is only used in the transmit and receive queues. (The buffer list uses the length field as a count field to allow larger buffer pools). If an attempt is made to use more than 28 buffer descriptors for a single received packet, an Excess Buffer Descriptor error is generated.

The transmit queue uses the frame options field to set transmit characteristics for individual packets:

- 10000 = Big endian byte ordering.
- 01000 = Interrupt after transmitting.
- 00100 = No CRC appended.
- 00010 = No PAD bytes, if short frame.

These bits can be combined to ask for combinations of characteristics. For example, 01110 would mean (1) little endian, (2) interrupt after transmission, (3) do not append CRC, and (4) do not pad a short packet.

Per-packet big endian controls can be useful in a hub application, where packets are received for transmission from a mixture of big and little endian sources. For computer applications, it is easier to use the global big endian control bit described in "DMA Control Register Bit Assignments (00h)" on page 17.

If the global packing enable bit (EnPack) described in "DMA Receive Fragment Size Register (14h)" on page 18 has not been set, the receive buffer list queue uses the frame options field to control packing and little endian or big endian data order.

- 10000 = Big endian byte ordering.
- 00001 = Enable buffer packing for the buffers in this frame, ignore the global enable bit.

If packing is enabled, the "DMA Receive Fragment Size Register (14h)" on page 18 controls the packing algorithm.

Buffer Descriptors

Each buffer descriptor has a pointer to the data buffer, control and status bytes, and a 2-byte-length field. When buffers are on the buffer list queue, the BuffData field points to the beginning of the buffer, and the BuffLength field reflects the allocated size of the unused buffer. When buffers are in use on the transmit and receive queues, the BuffData field points to the beginning of data, and the BuffLength field reflects the length of the data. The first table below shows the layout of a buffer descriptor.

Byte 3	Byte 2	Byte 1	Byte 0	Offset
BuffData				00
BDCtl	BDStat	BuffLength		04

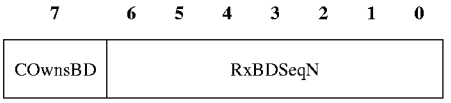
BuffData	Buffer Data Pointer	32-bit address of storage for bytes of data.
BDCtl	Buffer Descriptor Control	Control for this buffer descriptor.
BDStat	Buffer Descriptor Status	Status for this buffer descriptor.
BuffLength	Buffer Length	Length field for this buffer descriptor.

When buffers are placed in the buffer list queue, system software must set the length field to the allocated size.

As with frame descriptors, each queue makes slightly different use of the BDCtl and BDStat fields, as explained below.

Buffer Descriptor Control (BDCtl)

On the transmit queue, the BDCtl field is not used. On the receive queue, the BDCtl field holds the sequence number of buffer descriptors pointing into a single buffer area. The first buffer descriptor in the frame is numbered 0, the next 1, and so forth.



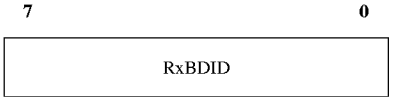
COwnsBD	Controller Owns Buffer Descriptor	=1. Controller owns the buffer descriptor. When the system sets COwnsBD, the buffer is free for reception. =0. System owns the buffer descriptor. When the controller clears COwnsBD, the buffer has been filled.
RxBDSeqN	Receive Buffer Descriptor Sequence Number	After reception: the sequence number for this buffer within the current buffer area.

On the free buffer list, the BDCtl field is used to record ownership of the buffer. This allows synchronizing allocation and freeing of buffer descriptors, to ensure that the controller does not “wrap around” and begin reusing buffers before the system can empty them.

Buffer Descriptor Status Field (BDStat)

On the transmit queue, the BDStat field is not used by the controller.

On the receive queue, the BDStat field is used as a buffer ID, which is copied from the free buffer queue.



RxBDID	Receive Buffer Descriptor ID	The buffer ID value.

On the free buffer queue, the BDStat field is used to pass the buffer descriptor ID number to the controller.

Note: Buffer IDs can only be unique if there are at most 256 buffers in a single buffer pool.

Register Initialization

On power-up and reset, the MAC control and status registers are set to the default values as described in "MB86974 REGISTER SET" on page 12.

Transmit Collision Count, CAM Data, and EEPROM/ROM Buffer registers are not set on power-up or reset. The Transmit Collision Count Register is reset at the beginning of transmitting a new packet. The CAM memory should be initialized before enabling usage of the CAM.

MAC Register Access

MAC Register access is controlled through the PCI bus interface. For more information on register read and write access, including MAC command and status registers, see "Initializing the DMA and MAC" on page 41.

Special Register Clear Operations

The Missed Packet Error Count Register is cleared on read. This ensures synchronization with software drivers, which accumulate a total count.

The transmit and receive status registers are cleared at the beginning of the next packet. For this reason, the values read from the register interface may not be stable. The value of these registers is stored in the FDStatus field of the Frame Descriptor in memory, for each packet transmitted or received, and software should examine the status values found in the system data structures.

Transmitting a Frame

To transmit a frame, the Transmit Enable bit in the Transmit Control Register must be set and the Transmit halt Request bit must be zero. In addition, the halt immediate and halt request bits in the MAC control register must be cleared. These conditions are normally set after such DMA controller initialization has occurred as storing a valid frame descriptor address in the Transmit Frame Pointer Register. The MAC will then signal the DMA engine to transfer bytes to the MAC transmit FIFO. The DMA transmit controller then controls the transfer of bytes to the MAC transmit FIFO.

The MAC transmit block will start transmitting the data in the FIFO, but will retain the first 64 bytes until it has acquired the net. At that time, the MAC transmit block will request more data and transmit it until the DMA transmit controller signals the end of data to be transmitted. The MAC transmit block generates pad bytes, if needed, appends the calculated CRC to the end of the packet, and transmission ends. It sets the Completion bit in the Transmit Status Register, signaling the end of a transmission, which may in turn cause an interrupt.

Transmission of data across the MII interface is driven by either a 25 or 2.5 MHz MII transmit clock, TX_CLK. Transmission to the 10-Mbit/s endec is driven by a 10-MHz transmit clock, TXC_10.

The MAC transmit block does not begin transmission onto the net until there are 8 bytes of data in the MAC transmit FIFO. Since the first 8 bytes transmitted are the preamble and the Start Frame Delimiter, this gives an initial 16 byte times for DMA latency. The DMA transmit block does not begin transferring data to the MAC transmit FIFO, until either the entire packet is in the DMA RAM buffer, or the number of bytes in the DMA RAM buffer exceeds the DMA Transmit Threshold Register. If transmit underrun errors occur, the problem can be corrected by setting the DMA Transmit Threshold Register to a higher value.

The MAC transmit block will check the parity. If there is a parity error, the MAC transmit block aborts the transmission, resets the FIFO, and sets the MAC Parity Error bit in the Transmit Status Register.

IEEE 802.3 Transmit Protocols

The MAC transmit block consists of three state machines:

- The *main transmit* state machine implements the MAC-layer protocols and controls the other two machines.
- The *gap* state machine tracks and counts the inter-packet gap timing between packets.
- The *backoff* state machine implements the backoff and retry algorithm of the 802.3 CSMA/CD protocol.

Inter-Packet Gap (IPG) Timing

In half-duplex mode, the gap state machine is responsible for counting the 96 bit time inter-record gap from the de-assertion of the carrier sense signal. To precisely control the appropriate times for beginning transmission, it breaks the 96 bit times for inter-record gap into the first 64 bits and last 32 bits. If there is any traffic within the first 64-bit times, it resets the counter and resumes counting from zero. If there is traffic within the last 32 bits, it continues counting and signals the end at 96 bit times.

In full-duplex mode, the gap state machine starts counting at the end of transmission and signals the end at 96 bit times.

Collision Processing and Backoff

If the main transmit state machine detects a collision, it starts the backoff state machine counters and waits for the end of the backoff slot, before retransmitting the collision-causing packet again. Each time there is a collision for the same packet, the backoff state machine increments an internal attempt counter. Each backoff slot is a multiple (including zero) of 512 bit times. A subset of the value of an 11-bit pseudo-random number generator selects the multiple. The subset grows by 1 bit for each successive attempt. This implements the equation:

$$0 \leq r < 2^k$$
$$k = \min(n, 10)$$

where r is the number of slot times that the MAC has to wait in case of a collision, and n is the number of attempts. For example, after the first collision, n is 1 and r is a 1-bit wide random number of either 0 or 1. After the second attempt, r is a random number between 0 and 3; the state machine looks at the two least-significant bits of the generator (n = 2), which gives a value between 0 and 3.

To improve the statistical independence between two MACs using the same pseudo-random number generator, the MAC uses values from the CRC of previous successfully transmitted packets to modify the basic random number sequence.

The Transmit Operation

If data is to be transferred, if the inter-packet gap is OK, and if the MII is ready (there are no collisions, and if in full-duplex mode or if there is no Carrier Sense), the MAC transmit block:

1. Sends the preamble followed by the SFD.
2. Transmits 64 bytes of data regardless of the packet length, unless short transmission is enabled. This means that if the frame is less than 64 bytes, the transmit block pads the LLC data field with zeroes, and appends the CRC at the end of the packet, if CRC generation is enabled.
3. If there is any collision during this first 72 bytes (8 bytes of preamble and SFD and 64 bytes of the frame), the transmit block stops the transmission and transmits a jam pattern (32 bits of all 1s), increments the collision attempt counter and returns control to the Backoff State Machine, and retransmits the packet when the backoff time has elapsed and the gap time is OK.
4. If there are no collisions, the MAC transmit block transmits the rest of the packet, allowing the DMA engine to over-write this frame data. The transmit block then appends the CRC to the end. A FIFO underrun, an excessive deferral (Carrier Sense if in half-duplex mode), or more than 16 collisions causes the state machine to abort the packet (no retry) and prepare for the next packet in the queue.

In case of any transmission errors, the MAC transmit block sets the appropriate error bit in the Transmit Status Register. It might generate an interrupt, depending on the Transmit Control Register.

The Receive Operation

To receive a frame, the Receive Enable bit in the Receive Control Register must be set, and the Receive-Halt-Request bit must be zero. In addition, the halt immediate and halt request bits in the MAC Control Register must be cleared. These conditions are normally set after the DMA controller is initialized, storing a valid address into the Buffer List Frame Pointer Register, and initializing the Free Descriptor Area Base and Limit registers. For more information on initializing the data structures to enable reception, see "Receiving a Frame" on page 43.

The MAC receive block, when enabled, constantly monitors a data stream coming from either the MII or the optional external 10-Mbit/s endec. If the MAC is in loopback mode, the data stream comes from the MAC transmit block via the MII or 10Mbit/s endec lines.

The MAC receive block:

1. Searches for zero to 7 bytes of preamble, followed by the SFD, checking that the preceeding nibbles received are preamble. If the SFD is not the first non-preamble byte, the MAC receive block treats the packet as a fragment and discards it.
2. Combines subsequent nibbles (from the MII interface) or bits (from the 10Mbit/s endec lines) into bytes, generates parity, and CRC, and stores the byte with its parity in the MAC receive FIFO.
3. Signals the DMA receive block that data is present.
4. Receives the destination address. The first nibble of destination address follows the SFD. The CAM block attempts to recognize the destination address. If the CAM block rejects the address, the MAC receive block signals this condition. The DMA receive block then discards the data frame.
5. If during frame reception, the PHY asserts both RX_DV and RX_ER, the MAC receive block reports a CRC error for the current packet.

The DMA receive controller reads bytes from the MAC receive FIFO, checks parity, and moves the data into the DMA receive FIFO. When the MAC receive FIFO becomes empty, or when it drives out the last byte of a packet, the MAC receive block signals these conditions.

CAM Operation

To read or write the CAM, system software should first set the CAM Address Register, then read or write the CAM Data Register. All bytes are written, without regard to partial word enables. When writing the upper or lower 2 bytes of a double word, the driver software must also correctly write the adjacent 2-byte field. The controller does not support read/modify/write cycles to its internal DMA RAM.

The following diagram shows how the MAC reads CAM entries from the CAM memory. Entries are assumed to be in big endian order: #0-0 is the first byte of the first entry, #0-5 is the sixth and last byte of the first entry, and so on. There are 2 bytes after CAM entry #20: Rsv-2 and Rsv-3. There are also double words: MC#1, MC#2, and MC#3. These words are not used in CAM operation, but they are used in generating MAC control frames, as explained in "Remote Pause Operation" on page 38.

CAM Memory Map

Byte 3	Byte 2	Byte 1	Byte 0	
#0-0	#0-1	#0-2	#0-3	00
#0-4	#0-5	#1-0	#1-1	04
#1-2	#1-3	#1-4	#1-5	08
#2-0	#2-1	#2-2	#2-3	0C
#2-4	#2-5	#3-0	#3-1	10
#3-2	#3-3	#3-4	#3-5	14
#4-0	#4-1	#4-2	#4-3	18
#4-4	#4-5	#5-0	#5-1	1C
#5-2	#5-3	#5-4	#5-5	20
				...
#18-0	#18-1	#18-2	#18-3	6C
#18-4	#18-5	#19-0	#19-1	70
#19-2	#19-3	#19-4	#19-5	74
#20-0	#20-1	#20-2	#20-3	78
#20-4	#20-5	Rsv-2	Rsv-3	7C
MC#1-0	MC#1-1	MC#1-2	MC#1-3	80
MC#2-0	MC#2-1	MC#2-2	MC#2-3	84
MC#3-0	MC#3-1	MC#3-2	MC#3-3	88

Transmit Pause Operation

To enable full duplex PAUSE operation, the special broadcast address for MAC control packets must be programmed into the CAM, and the corresponding CAM enable bit set. This can be any CAM location. "Remote Pause Operation" below specifies how some specific CAM locations may be preferred, to optimize CAM entry utilization.

The MAC receive circuit recognizes the full duplex PAUSE operation when the following conditions are met:

- The type/length field has the special value for MAC control packets, 0x8808.
- The CAM recognizes the packet.
- The length of the packet is 64 bytes.
- The operation field specifies PAUSE operation.

When a full duplex PAUSE operation is recognized, the MAC receive circuit loads the operand value into the Pause Count Register and signals both the MAC and the DMA engine that pause should begin at the end of the current packet, if any.

The pause circuit maintains the pause counter and decrements it to zero. It then signals the end of the PAUSE operation and allows the DMA transmit circuit to resume.

If a second full duplex PAUSE operation is recognized while the first operation is in effect, the pause counter is reset with the current operand value. Note that a value of 0 can cause a pause operation in progress to stop prematurely.

Remote Pause Operation

The MB86974 Flow Control 100/10-Mbit/s Ethernet MAC supports full programmability of MAC control frames to support both PAUSE operation and future uses of MAC control.

To send a remote PAUSE operation or other MAC control frame, follow these steps:

1. Program CAM location #0 with the destination address.
2. Program CAM location #1 with the source address.
3. Program CAM location #20 with the MAC control type-field, PAUSE operation opcode, and operand value. The 2 reserved bytes after CAM location #20 should be written with 0000h.
4. Program the three double-word locations MC#1, MC#2, and MC#3 with 0000_0000h.
5. Write the Transmit Control Register, setting the SdPause bit.

The destination address and source address are normally the special broadcast address for MAC control frames and the local station address, respectively. These CAM entries can be enabled for use in address filtering. CAM entry #20 should not be enabled, when used as part of flow control transmission.

Upon completion, the transmit status is written to the Transmit Control Frame Status Register. The DMA engine generates an interrupt if the Transmit Control Complete Enable bit (10) of the Interrupt Enable Control Register is set.

Error Signaling

The error and abnormal operation flags set by the MAC are arranged into transmit and receive groups, and can be found in either the Transmit Status Register (Tx_stat, page 26) or the Receive Status Register (Rx_stat, page 28). In addition, the Missed Error Count Register (page 30) counts packets missed for system network management purposes. Refer also to "MB86974 REGISTER SET" on page 12 for the formats of the flags and counters.

Reporting of Errors in Transmit

Transmit operation terminates when the entire packet (preamble, SFD, data, and CRC) has been successfully transmitted to the physical medium without encountering a collision, or alternatively aborted. In addition, the MAC transmit block detects and reports both internal and network errors.

Under the conditions listed in the following table, transmission is aborted and a status bit is set. Many of the set bits can also generate interrupts, if the corresponding Interrupt Enable bit has been set in the Transmit Control Register (page 25).

Reporting of Errors in Transmit

MAC Transmit Parity Error	A parity bit protects data coming from the DMA transmit controller via the DII into the MAC transmit FIFO. A parity error sets the TxParErr bit of the Transmit Status Register and halts the transmitter, if interrupt is enabled.
MAC Transmit FIFO Underrun	The 80-byte MAC transmit FIFO can handle a worst-case DMA latency of 1.28µs (128 bit times, or 16 byte times), because 64 bytes are retained for possible retransmission after a collision. A MAC transmit FIFO underrun usually indicates a PCI bus-latency problem, since the DMA transmit controller has more than enough bandwidth to keep up. Such an underrun sets the Underrun bit in the Transmit Status Register.
Lost Carrier	Carrier Sense (CrS) is monitored from the beginning of the Start Frame Delimiter (SFD) to the last byte transmitted. A lost carrier condition indicates that CrS was never present or was dropped during transmission (a possible network problem), but transmission is **not** aborted. During loopback mode, Tx_en drives CrS. During full-duplex operation, CrS is not passed to the transmit block, and lost carrier is not asserted. Lost carrier sets the LostCrS bit in the Transmit Status Register.
Excessive Collision	Whenever the MAC encounters a collision during transmit, it backs off, updates the collision counter, and tries again later. When the counter equals 16 (16 attempts resulted in a collision), transmission is aborted. Excessive collisions probably indicate a network problem. Excessive collision sets the ExColl bit in the Transmit Status Register.
Late Collision (Transmit Out-Of-Window Collision)	In a correctly operating network, the controller sees a collision (if there is one) within the first 64 bytes of data being transmitted. If a collision occurs after this time, a possible network problem is detected. Late collision sets the LateColl bit in the Transmit Status Register, and transmission of the packet is aborted; that is, late collisions are not retried.
Signal Quality Error (SQE)	In 10 Mb/s mode, the MAC checks for a "heartbeat" at the end of a transmitted packet. This is a short collision signal within the first 40 bit times after end of transmission. SQE sets the SQErr bit in the Transmit Status Register.
Deferral	During any attempt to send a packet, the MAC may have to defer the transmission because of a pre-occupied network. This is not an error; it is used as a network activity indicator, but only when collisions do not occur. Deferral sets the TxDeferred bit of the Transmit Status Register.
Excessive Deferral	During the first attempt to send a packet, the MAC may have to defer the transmission because of a pre-occupied network. If the deferral time is longer than two maximum-sized packet times (2.4288ms for either of the 10-Mbit/s operation modes or 0.24288ms for the 100-Mbit/s operation mode), transmission is aborted. Excessive deferral indicates a possible network problem. Excessive deferral sets the ExDefer bit of the Transmit Status Register.
Paused	During any attempt to send a packet, the MAC may have to defer the transmission because the Transmitter has been paused by the reception of MAC Control packet containing a PAUSE Operation. This is not an error, but is used as a network activity indicator. To assist software in marking packets which experience a pause, the Paused bit is set on the last packet before a PAUSE will take effect.

Reporting of Errors in Receive

The MAC receive block starts putting received data from the physical medium into the MAC receive FIFO after detecting the SFD. It also checks for MAC receive FIFO overflow during reception. At the end of reception, the MAC receive block looks for external errors (Alignment, CRC, and Frame Too Long).

MAC Receive Parity Error	A parity bit protects data once it enters the MAC receive FIFO. A parity error sets the RxParErr bit of the Receive Status Register and halts the receiver if interrupt is enabled.
Alignment Error	At the end of reception, the MAC receive block checks that the incoming packet has been correctly framed on an 8-bit boundary. If it is not, and the CRC is invalid, data has been disrupted through the network, and the MAC receive block reports an alignment error. A CRC error is also reported. The AlignErr bit and the CRCErr bits are set in the Receive Status Register.
CRC Error	At the end of reception, the MAC receive block checks the CRC for validity and reports a CRC error if it is invalid. CRC, frame alignment, and length errors are the network errors detected by the receive unit. They might be detected in the following combinations: CRC error only Frame alignment and CRC errors only Length and CRC errors only Frame alignment, length, and CRC errors
Overflow Error	During reception, incoming data is put into the MAC receive FIFO before it is transferred to the DMA receive controller. If the MAC receive FIFO fills up because of excessive system latency or other reasons, the MAC receive block sets the Overflow Error bit of the Receive Status Register.
Long Error	The MAC receive block checks the length of the incoming packet at the end of reception. If the length is longer than the maximum frame size of 1518 bytes, the MAC receive block reports receiving a long error, unless long frame mode is enabled.
MII Error	The PHY informs the MAC if it detects a media error (such as coding violation) by asserting Rx_er. When the MAC sees Rx_er asserted, it rejects the received packet. A CRC error is forced and the packet is terminated. An alignment error or short error could then also be detected.

Accessing Station Management Data

To access the station management data:

1. System software reads the Busy bit to ensure that the MD is not busy.
2. For a write operation, the data is written into the Data Register before setting the Control Register.
3. Software writes the MDC address and the read/write flag, and then sets the Busy bit.
4. The controller completes the operation and clears the Busy bit.
5. For a read operation, when system software detects that the Busy bit is cleared, it reads the Data Register.

Accessing an EEPROM or ROM

To access an external EEPROM or serial ROM:

1. System software reads the Busy bit to ensure that the EEPROM driver is not busy.
2. For a write operation, the data should be written into the Data Register before setting the Control Register.
3. Software writes the address and the read/write flag, then sets the Busy bit.
4. The controller completes the operation and clears the Busy bit.
5. For a read operation, when system software detects that the Busy bit is cleared, it reads the Data Register.

10 Mb/s Operation over 7-Wire Interface

The MB86974 controller implements an optional 7-wire interface. This interface is not an approved standard, and several different vendors have implemented slightly different signaling conventions. The current design supports what is known as "National Semiconductor" mode. The timing diagrams for operation over the 7-wire interface are on page 57.

Software Initialization

This section gives information on programming the PCI-based 100/10-Mbit/s Ethernet controller. Programming details differ slightly, depending on the control mode chosen; batch processing or continuous polling. The descriptions here assume continuous polling, unless otherwise noted.

The topics covered include:

- Initializing the PCI Interace
- Initializing the DMA and MAC
- Queue Initialization
- Transmitting a Frame
- Finding Out if Transmission is Complete
- Receiving a Frame
- Processing Received Frame Descriptors
- Freeing Buffers
- Processing Interrupts

Initializing the PCI Interface

At system initialization, the IDSEL signal can be used to write the PCI configuration registers. This is normally done by: (1) having the system map the controller into a start-up memory address space; and (2) having data transfers to those addresses generate the IDSEL signal.

Normally, some registers can be initialized from data in an optional EEPROM or ROM as described in "Accessing an EEPROM or ROM" on page 40. This requires setting a Base Address Register first.

Registers that *must be* initialized include:

- PCI I/O or Memory Base Address, to map registers into I/O or memory space
- PCI Command, to customize PCI capabilities

A register that *might be* initialized includes PCI Interrupt, to customize latency or to route interrupts.

Initializing the DMA and MAC

After PCI initialization, the DMA and MAC control registers are normally mapped into I/O space, or memory space. They can be read or written from the mapped space. Again, these registers can be initialized from data in an optional EEPROM or ROM, as described in "Accessing an EEPROM or ROM" on page 40.

Registers that *must be* initialized include:

- DMA Transmit Frame Pointer (to initiate transmission)
- DMA Buffer List Frame Pointer (to provide buffers for reception)
- DMA Free Descriptor Area Base and Limit (to initialize the receive notification area)
- DMA Transmit Polling Count (to customize polling for packets to transmit)
- DMA Transmit Threshold (to customize handling of transmit latency)
- MAC Transmit Control (to change default transmission settings)
- MAC Receive Control (to change default reception settings)
- MAC CAM Control (to customize station and multicast-group address recognition)
- MAC CAM Address and Data (to provide station address and other address filtering)
- MAC CAM Enable (to enable individual CAM entries after setup)

Registers that *might be* initialized include:

- MAC Control (to customize MAC configuration)
- DMA Transmit Burst Size (to customize transfer sizes)

Queue Initialization

Before starting the controller, the system needs to set up the transmit queue, the buffer list queue, and the receive descriptor area.

There are two modes of operation for the transmitter; batch processing and continuous polling.

For *batch processing*, the system software sets up a linked list of frame descriptors to send, with the last frame descriptor containing an EOL indicator. When the last frame descriptor is transmitted, the Transmit Frame Pointer Register loads the EOL indicator, and transmission terminates. Later, the system must restart transmission by storing a new value in this register.

For *continuous polling*, the system software sets up a linked list of frame descriptors to transmit, which ends with a dummy frame descriptor. The linked list may be initially empty, except for the dummy frame descriptor. The system owns the dummy frame descriptor to prevent the controller from accessing it. When a new packet is to be sent, the dummy frame descriptor is overwritten, as described in "Transmitting a Frame" below.

Initializing the Buffer List

The buffer list queue is initialized by setting up a linked list of frame descriptors with one or more frame descriptors, each containing a list of free buffer descriptors. The list can be any one of the following:

- A single frame descriptor containing a large number of free buffer descriptors
- A linked list of frame descriptors
- A circular queue, in which the last frame descriptor points to the first frame descriptor

For the first two, the FDNext field would have the EOL bit set; for the third, the FDNext field of the last frame descriptor would point to the first frame descriptor. The Receive Buffer Fragment Size Register can be set to globally enable packed buffer usage. Alternatively, the frame descriptor control field (FDCtl) can be used to select packed or unpacked buffer usage on a per buffer-area basis. For packed buffer usage, the buffer ID fields can be set to assist in memory management. For more details on enabling buffer packing, see "DMA Receive Fragment Size Register (14h)" on page 18, "Free Descriptor Area Base Register (1Ch)" on page 19, and "Buffer Descriptors" on page 34.

Initializing the Receive Descriptor Area

The receive descriptor area is initialized by writing the Descriptor Area Base and Limit registers. The controller uses these registers to initiate writing of the receive queue in the receive descriptor area.

Transmitting a Frame

"Initializing the Transmit Queue" above describes batch processing transmission. For each batch of frames to send, the system initializes the transmit queue and writes the head of the queue into the Transmit Frame Pointer Register.

For continuous polling transmission, a dummy frame descriptor owned by the system terminates the list of frame descriptors. When the controller reaches the dummy record, it enters a polling mode. In this mode, the controller periodically reads the frame descriptor control (FDCtl) field and waits for the system to clear the FDOwner bit. The Transmit Polling Counter Register controls the frequency of polling.

To transmit a frame in continuous polling mode, the system writes a new frame descriptor for the frame to send at the end of the transmit queue. To do this, the system: (1) overwrites the old dummy frame descriptor; (2) creates a new dummy frame descriptor; (3) sets the next field of the old frame descriptor to the new dummy frame descriptor; and (4) clears the FDOwner bit of the old frame descriptor, thus giving ownership to the controller.

Finding Out if Transmission is Complete

To find out if a transmission is complete, the system can:

- Request an interrupt
- Poll the FDCtl field of transmitted frame descriptors, to determine system ownership
- Poll the Transmit Frame Pointer Register

Interrupts can be requested at the end of each frame sent, or at the end of selected frames. When polling the Transmit Frame Pointer Register, the system can look for an invalid value (batch processing mode) or can look for the address of the dummy frame descriptor (continuous polling mode).

Receiving a Frame

To permit the MAC to receive frames, system software must:

- Initialize the free buffer list and free descriptor areas, as described in "Queue Initialization" on page 42.
- Write a dummy frame descriptor into the free descriptor area, and set the FDOwner bit of the FDCtl field so the controller owns it.
- Initialize the Receive Frame Pointer Register to the address of the dummy frame descriptor in the free descriptor area.

To notify system software that frames are received, system software must:

- Request an interrupt for each frame received
- Poll the dummy frame descriptor and look for the FDOwner bit to be set

To enable interrupt, the Completion Interrupt Enable bit of the Receive Control Register must be set.

After a frame is received, the system must

- Process the frame descriptor and free it for reuse at a future time
- Free buffers as they are returned and add them to the free buffer list

Processing Received Frame Descriptors

The free descriptor area is used in a FIFO manner, but different applications take different amounts of time to process frames and return associated buffers. Frame descriptors that the controller allocates are therefore copied to another area and freed in the order in which they are received. They are then passed up the protocol stack.

Freeing Buffers

Buffers can be allocated in two ways:

- By starting a new frame in a new buffer
- By placing several frames or parts of frames in a single buffer

Either the Buffer Fragment Size Register or the frame descriptor control field controls the allocation mode. The single frame mode has simpler memory management but less-efficient memory utilization. Just the opposite is true of the packed buffer mode.

Because having multiple frames or fragments of frames in the same buffer area is possible, packed buffers require additional managing. The controller counts the number of buffers created in the same buffer area and provides this count as the RxBDSeqN field of the BDCtl field in the buffer descriptor. System software can then count returned fragments, until all fragments are returned.

A buffer ID value, RxBDID, is copied from the buffer descriptors in the free buffer queue to the buffer descriptors in the received frame queue. Up to 256 ID values are available. If more are needed, ID extension bits can be provided in the FDSystem field or high order bits from the buffer pointer values, which point into the buffer, can be calculated.

Processing Interrupts

An interrupt generally occurs on a shared interrupt line. To see if this PCI device is the source of an interrupt, system software reads the Interrupt Source Register. Based on the register's contents, system software may need to read additional registers, such as the Transmit or Receive Status registers.

MB86974 PIN Descriptions

The MB86974 comes in a 144-pin PQFP package. The following table lists the pin assignments and defines them:

Pin Name	Pin Number	Pin Type	Description
$\overline{\text{FRAME}}$	21	PCI_IO	PCI cycle frame, asserted by the current master (Active LO)
$\overline{\text{STOP}}$	27	PCI_IO	Target requests Master to Terminate or Abort cycle (Active LO)
$\overline{\text{TRDY}}$	23	PCI_IO	Target Ready (Active LO)
$\overline{\text{IRDY}}$	22	PCI_IO	Initiator Ready (Active LO)
IDSEL	4	PCI_IO	Initialization Device Select
$\overline{\text{DEVSEL}}$	24	PCI_IO	Target Device Selected (Active LO)
AD[31:0]	133 135:136 139:140 142:143 1 6 8:9 12:14 16:17 34:35 37:39 42:44 47:48 50:51 53:54 57:58	PCI_IO	PCI address and data bus. Carries address information during the address phase and data during the data phase.
$\overline{\text{C\_BE}}$[3:0]	3,20,31,45	PCI_IO	PCI Command and Byte Enable (Active LO)
PAR	30	PCI_IO	Parity. Even parity for AD[31:0] & $\overline{\text{C\_BE}}$[3:0]
$\overline{\text{GNT}}$	131	PCI_IO	Grant Bus (Active LO)
$\overline{\text{REQ}}$	132	PCI_IO	Request Bus (Active LO)
$\overline{\text{PERR}}$	28	PCI_IO	Parity Error (Active LO)
$\overline{\text{SERR}}$	29	PCI_IO	System Error (Active LO)
$\overline{\text{INTA}}$	126	PCI_O	Requests an Interrupt (Active LO, Wire AND)
PCI_CLK	128	PCI_I	16-33 MHz PCI Clock
$\overline{\text{RST}}$	127	PCI_I	Reset (Active LO)
COL	95	MII_I	Collision Detect. This input from the PHY reflects a collision on the media when in half-duplex mode. In full-duplex mode, COL should be always low.
CRS	94	MII_I	Carrier Sense is asserted by the PHY whenever it detects that either the transmit or receive lines is in active in half-duplex mode. In full-duplex mode and repeater mode, CRS is asserted only when the receive line is active. CRS is not synchronous to either the transmit or receive clocks.
MDC	115	MII_O	Management Data Clock. This clock is driven by the MB86974 during management information transfers between the MB86974 and the serial Management Data port on a PHY.
MDIO	116	MII_I/O	Management Data Input/Output. The MDIO signal is used to transfer both control and status information. During the transfer of control information, MDIO functions as an output. During the transfer of status information, MDIO functions as an input that driven by the PHY. All transfers are synchronized with the MDC clock.
RX_CLK	106	MII_I	Receive Clock. This 25 MHz or 2.5 MHz signal from the PHY is used to synchronize receive data.

Pin Name	Pin Number	Pin Type	Description
RX_D[3:0]	114,113 111,110	MII_I	Receive Data. These four inputs are used to transfer receive data and are connected to the receive data outputs of the PHY.
RX_DV	109	MII_I	Receive Data Valid. The assertion of RX_DV by the PHY indicates that a 4-bit data nibble is being presented on the RX_D[3:0] lines. This operation is performed synchronous to the receive clock.
RX_ER	105	O	Receive Error. The receive error output is asserted by the PHY synchronous to the receive clock and indicates that an error occurred during the receive operation.
TX_CLK	103	MII_I	Transmit Clock. This 25 MHz or 2.5 MHz signal from the PHY drives transmit data.
TX_D[3:0]	96:99	MII_O	Transmit Data. The 4-bit transmit value contains data to be transmitted on to the network. TX_D[3:0] are connected directly to the transmit data inputs of the PHY.
TX_EN	102	MII_O	Transmit Enable. Assertion of TX_EN indicates that valid transmit data has been driven onto the bus by the MAC. At the same time that TX_EN is asserted, the PHY should begin reading data on TX_D[3:0]. Data transmission on TX_D[3:0] is terminated when TX_EN is negated.
TX_ER	104	MII_O	Transmit Error. The assertion of TX_ER by the controller indicates that an error has occurred during a data transmit operation. TX_ER is asserted synchronous to TX_CLK.
COL_10	88	7-W_I	Collision Detect on 7-wire interface
TXC_10	91	7-W_I	Transmit Clock on 7-wire interface
TXD_10	90	7-W_O	Transmit Data on 7-wire interface
TXEN_10	89	7-W_O	Transmit Enable on 7-wire interface
CRS_10	82	7-W_I	Carrier Sense on 7-wire interface
RXC_10	81	7-W_I	Receive Clock on 7-wire interface
RXD_10	83	7-W_I	Receive Data on 7-wire interface
LOOP_10	84	7-W_O	Loopback Control on 7-wire interface
LINK_10	87	7-W_I	LINK Detect on 7-wire interface
PROM_DI	75	STD_I	EEPROM/ ROM data input
PROM_DO	76	STD_O	EEPROM/ ROM data output
PROM_CLK	77	STD_O	EEPROM/ ROM clock
PROM_CS	78	STD_O	EEPROM/ ROM chip select
CAM_LOAD	72	STD_O	External CAM address load
$\overline{\text{CAM\_HIT}}$	71	STD_I	External CAM hit
JTCK	123	STD_I	JTAG Test Clock
JTMS	119	STD_I	JTAG Test mode select
JTDI	122	STD_I	JTAG Test data in
JTDO	117	STD_O	JTAG Test data out
$\overline{\text{JTRST}}$	118	STD_I	JTAG Test reset
SE	59	STD_I	SCAN Enable
SI[2:1]	66,62	STD_I	SCAN chain inputs
SO[2:1]	67,63	STD_O	SCAN chain outputs
STM	68	STD_I	SCAN Test Mode

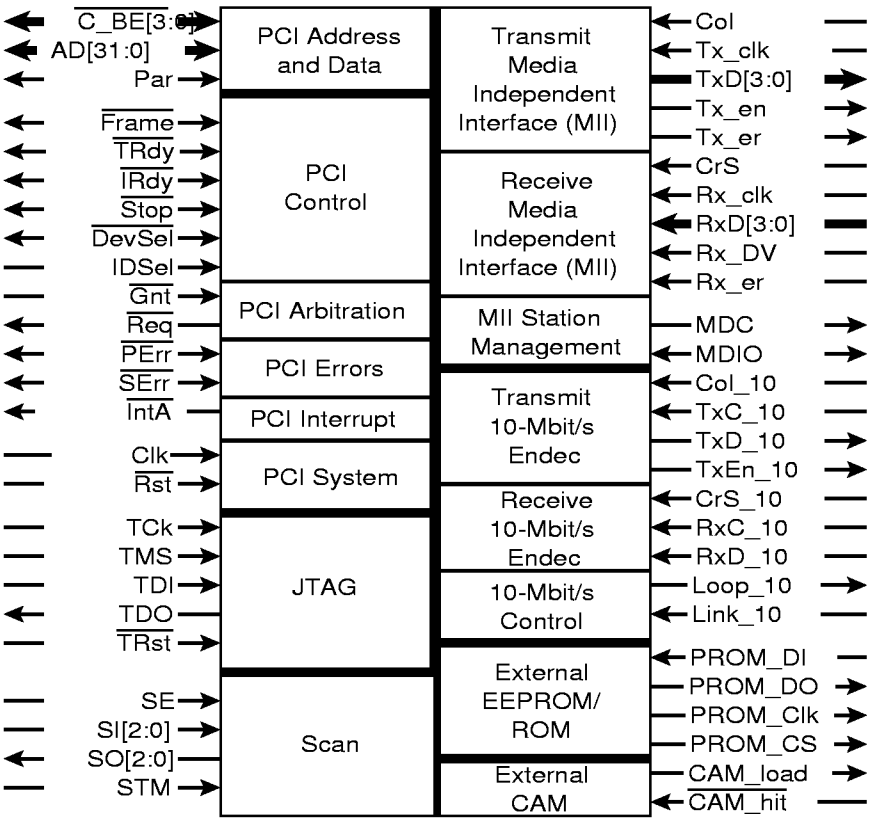
Pin Name	Pin Number	Pin Type	Description
VDD	5 11 18 25 32 41 56 61 65 70 73 80 86 92 101 107 120 125 130 137	PWR	3.3V power connections.
VSS	2 7 10 15 19 26 33 36 40 46 49 52 55 60 64 69 74 79 85 93 100 108 112 121 124 129 134 138 141 144	PWR	Ground connections.

Signal Pin Descriptions

The following diagram shows all of the external signals for the Flow Control 100/10-Mbit/s Ethernet controller, divided into functional groups. Definitions of the signals are grouped in a functional area, that gives each signal's symbolic name, full name, direction and clock domain. The groups are:

- 13 for PCI control, arbitration, error reporting, interrupt, and system
- 37 for PCI address and data
- 18 for MII transmit and receive
- 9 for an optional 10-Mbit/s Manchester encoder-decoder (endec)
- 4 for an optional external EEPROM/ROM
- 2 for an optional external CAM interface
- 5 for JTAG
- 8 for internal scan

Pin Signals



PCI Signals

MB86974 includes all 50 PCI signals for the Flow Control 100/ 10-Mbit/s Ethernet controller. These include the 49 signals required for a PCI master, plus the $\overline{\text{INTA}}$ (Interrupt A) signal. The part drives sustained tri-state signals high for one clock before returning them to the high-impedance state. Signals with names with overlines are active low signals. All signals are referenced to the rising edge of PCI_CLK, except $\overline{\text{SERR}}$, $\overline{\text{RST}}$, and $\overline{\text{INTA}}$. The $\overline{\text{SERR}}$ signal is asserted synchronous, but is deasserted asynchronous to the PCI_CLK.

PCI Control Signals

FRAME

Cycle frame. Driven by the current master to signal how long an access lasts. In a bus transaction terminated by the master, the master asserts FRAME beginning with the first or address cycle, and holds it until the next to last cycle, when it deasserts it. (See the STOP signal for a description of target abort.)

TRDY

Target ready. During a read, the target asserts TRDY to indicate that it is driving valid data on AD[31:0]. During a write, the target asserts TRDY to indicate it is ready to receive data over AD[31:0]. A data transfer occurs in a cycle when both TRDY and IRDY are asserted.

STOP

The target asserts STOP to request that the master terminate or abort the current transaction.

DEVSEL

Device select. Driven by the target to confirm that it has decoded the address as referring to itself.

IDSEL

Initialization device select. Equivalent to chip select, IDSEL is a fully decoded addressing mechanism for configuring read and write transactions.

IRDY

Initiator ready. IRDY. During a read, the initiator asserts IRDY to indicate that it is ready to receive data on AD[31:0]. During a write, the initiator asserts IRDY to indicate it is driving valid data over AD[31:0]. A data transfer occurs in a cycle when both TRDY and IRDY are asserted.

PCI Address and Data Signals

AD[31:0]

Address and data bus. Carries address information during the address phase and data during the data phase.

C_BE[3:0]

Command and byte enable. During the address phase, C_BE[3:0] defines the type of transaction. During the data phase, it indicates the validity of bytes carried on AD[31:0]. The tables below show the correspondence between the byte-enable signals and bytes on the address and data bus during the data phase.

Correspondence Between C_BE[3:0] and AD[31:0] (Little-Endian Byte Order)

Address and data bus AD[31:0]																													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Byte 3 (MSB)								**Byte 2**								**Byte 1**								**Byte 0 (LSB)**							
C_BE[3]								C_BE[2]								C_BE[1]								C_BE[0]							

Correspondence Between C_BE[3:0] and AD[31:0] (Big-Endian Byte Order)

Address and data bus AD[31:0]																													
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Byte 0 (MSB)								**Byte 1**								**Byte 2**								**Byte 3 (LSB)**							
C_BE[3]								C_BE[2]								C_BE[1]								C_BE[0]							

PAR

Parity. Provides even parity for the information carried on AD[31:0] and $\overline{C\_BE}$[3:0] by causing AD[31:0], $\overline{C\_BE}$[3:0], and Par to contain an even number of bits equal to one. The Par signal is active during the clock period after AD and $\overline{C\_BE}$ contain valid data.

PCI Bus Arbitration Signals

$\overline{GNT}$

Grant. A point-to-point signal from the arbiter to an agent, signaling to the agent that the arbiter has granted it bus ownership. The PCI specification specifies use of a standard tri-state driver.

$\overline{REQ}$

Request. A point-to-point signal from an agent to the arbiter, signaling to the arbiter that the agent desires bus ownership. In normal operation, $\overline{REQ}$ is an output, but during reset it enters a high-impedance state.

PCI Error Signals

$\overline{PERR}$

Parity error. Kept at a high-impedance state, unless an agent receives non-special cycle data with a parity error. In such a case, it asserts $\overline{PERR}$ two clocks later and then, as with all sustained tri-state signals (unlike $\overline{SERR}$), drives it high for one clock before returning to the high-impedance state.

$\overline{SERR}$

System error. The $\overline{SERR}$ signal is used to signal both parity errors during an address cycle of a bus transaction and all errors other than data parity errors.

PCI Interrupt Signal

$\overline{INTA}$

Interrupt A. Requests an interrupt.

PCI System Signals

PCI_CLK

Clock. All PCI inputs are sampled on the rising edge of clock. The controller supports 100-Mb/s Ethernet when operating at 16-33 MHz frequency operation. May be held low at any time, to conserve power. May only be stopped in a low state.

$\overline{RST}$

Reset. Causes all output signals to enter a high-impedance state, and clears all registers. It does not affect on-chip RAM or FIFOs. Upon deassertion, the software drivers are responsible to check for the presence of a serial EEPROM/ROM, and if present, to read in the station address, and other configuration parameters.

MII Signals

The MII is the interface between the Flow Control 100/10-Mbit/s Ethernet controller and the PHY. For a detailed description of these signals, see the MII sections of the 802.3u documents.

COL

Collision. Asserted asynchronously with minimum delay from the start of a collision on the medium.

TX_CLK

Transmit clock, from PHY. The controller drives TX_D[3:0] and TX_EN off the rising edge of the TX_CLK. The PHY samples them on the rising edge of the TX_CLK.

TX_D[3:0]

Transmit data. Transmit data is aligned on nibble boundaries. TX_D[0] corresponds to the first bit to transmit on the physical medium and is the LSB of the first byte, followed by the fifth bit of that byte during the next clock.

TX_EN

Transmit enable. TX_EN provides precise framing for the data carried on TX_D[3:0]. It is active during the clock periods that TX_D[3:0] contains valid data to be transmitted from preamble through CRC.

TX_ER

Transmit coding error. TX_ER is driven synchronously to TX_CLK and is sampled continuously by the PHY entity. If asserted for one or more TX_CLK periods, it causes the PHY to emit one or more symbols that are not part of the valid data or delimiter set somewhere in the frame being transmitted.

CRS

Carrier sense. Asserted asynchronously with minimum delay from the detection of a non-idle medium.

RX_CLK

Receive clock, from PHY. RX_CLK is a continuous clock. Its frequency is 25 MHz for 100-Mbit/s operation and 2.5 MHz for 10-Mbits/s. RX_D[3:0], RX_DV, and RX_ER are driven by the PHY from the falling edge of RX_CLK and sampled on the rising edge of RX_CLK by the MAC.

RX_D[3:0]

Receive data. RX_D is aligned on nibble boundaries. RX_D[0] corresponds to the first bit received on the physical medium, which is the LSB of the byte in one clock period and the fifth bit of that byte in the next clock.

RX_DV

Receive data valid. PHY asserts RX_DV synchronously and holds it active during the clock periods that RX_D[3:0] contains valid received data. PHY asserts RX_DV no later than the clock period when it places the first nibble of the SFD on RX_D[3:0]. If PHY asserts RX_DV prior to the first nibble of the SFD, RX_D[3:0] carries valid preamble symbols.

RX_ER

Receive error. PHY asserts RX_ER synchronously whenever it detects a physical medium error; for example, a coding violation. PHY asserts RX_ER only when it asserts RX_DV.

MII Station Management Signals

MDC

Management Data Clock. Timing reference for transfer of information on the MDIO signal. With the PCI clock at 33 MHz, the MDC clock has a maximum clock frequency of 33/14 = 2.36 MHz. The minimum clock period is 424 ns.

MDIO

Management Data I/O. Transfers data to or from a PHY attached to the MII. The controller can initiate a sequence to determine whether a PHY is attached if provided with a pull-down resister.

External 10-Mbit/s Endec Signals

These signals support connection to an optional 10BASE-T PHY. This mode of operation is distinct from the 10-Mbit/s operation mode of the MII. The external endec uses a 1-bit serial signal, but the MII supports a 4-bit parallel interface.

COL_10

Collision detect on 10-Mbit/s endec. Asserted when a 10-Mbit/s PHY detects a collision. Ignored if 10-Mbit/s PHY is not enabled by the Master Control Register. Tie low if interface not in use.

TXC_10

Transmit clock on 10-Mbit/s endec. Clock from 10-Mbit/s PHY to transfer data. Ignored if 10-Mbit/s PHY is not enabled by the Master Control Register. Tie low if interface not in use.

TXD_10

Transmit data on 10-Mbit/s endec. Data line for transmitting to the 10-Mbit/s PHY. Stays low if 10-Mbit/s PHY is not enabled by the Master Control Register.

TXEN_10

Transmit enable on 10-Mbit/s endec. Asserted by the controller when it is ready to transfer data. Stays low if 10-Mbit/s PHY is not enabled by the Master Control Register.

CRS_10

Carrier sense on 10-Mbit/s endec. Asserted when a 10-Mbit/s PHY has data to transfer. Ignored if 10-Mbit/s PHY is not enabled by the Master Control Register. Tie low if interface not in use.

RXC_10

Receive clock 10-Mbit/s endec. Clock from 10-Mbit/s PHY to receive data. Ignored if 10-Mbit/s PHY is not enabled by the Master Control Register. Tie low if interface not in use.

RXD_10

Receive data 10-Mbit/s endec. Data line for receiving from the 10-Mbit/s PHY. Ignored if 10-Mbit/s PHY is not enabled by the Master Control Register. Tie low if interface not in use.

LOOP_10

Loop back at 10-Mbit/s endec. Output signal driven by bit 7 of MAC Control Register.

LINK_10

Link status of 10-Mbit/s endec. Input signal to convey link status of the 10-Mbit/s encoder-decoder. Stored in bit 15 of the MAC Control Register. Ignored if 10-Mbit/s PHY is not enabled. Tie low if the interface is not in use.

External EEPROM or ROM Interface

These four signals control an external EEPROM or ROM. Use of these signals is optional.

PROM_DI

EEPROM/ROM data input. Data line for transmitting from external EEPROM/ROM to the controller. Must be high with no EEPROM present. An internal resistor pull-up is provided.

PROM_DO

EEPROM/ROM data output. Transfers data from the controller to an external EEPROM/ROM.

PROM_CLK

EEPROM/ROM clock. Clock for transmitting to and from an external EEPROM/ROM. With the PCI clock at 33 MHz, the maximum clock frequency for PROM_CLK is 33/34 = 0.97 MHz. This is compatible with the slowest commercial parts, which specify a maximum frequency of 1 MHz.

PROM_CS

EEPROM/ROM chip select. Used to frame transmissions to and from an external EEPROM/ROM.

External CAM Interface Signals

There are two external CAM interface signals. Use of these signals is optional.

CAM_LOAD

External CAM address load. Signals the external CAM to begin loading the destination address from RX_D[3:0]. This signal is optional even with an external CAM, because external circuitry can derive it from MII signals, just as the MAC does.

$\overline{\text{CAM\_HIT}}$

External CAM hit. Notification from the external CAM that it has recognized this packet's destination address. Held inactive by a pull-up resistor if the external CAM is not present.

JTAG Interface Signals

This group of five signals is used to implement the JTAG Boundary Scan standard for board-level testing of I/O connectivity of the devices.

JTCK

Test clock. JTAG clock used for synchronization of other JTAG signals and for clocking data in and out of the boundary scan chain. Tie low if not in use.

JTMS

Test mode select. Enables boundary scan using serial in/serial out ports. Sampled on rising edge of JTCK. Tie low if not in use.

JTDI

Test data in. Serial input port for clocking in test data to be shifted to the output at the end of the boundary scan chain (JTDO). Tie low if not in use.

JTDO

Test data out. Serial output port for clocking out test data shifted from the input at the beginning of the boundary scan chain (JTDI).

$\overline{\text{JTRST}}$

Test reset. Asynchronous reset to JTAG logic. Tie high if not in use.

Internal Scan Interface Signals

This group of eight signals is used to implement scan test vectors for automatic testing. The MB86974 Ethernet controller requires two scan chains. The Scan Test Mode signal is required to be asserted during scan testing, in order to hold some internal logic elements in a testable state. These include elements related to clock muxing, power management, software reset, and memory blocks.

SE

Scan Enable. This signal allows the shifting of data through the internal scan chains. Tie low if not in use.

SI[2:0]

Scan Data Inputs. Input bus provides serial input for data shifted into the internal scan chains. Tie low if not in use.

SO[2:0]

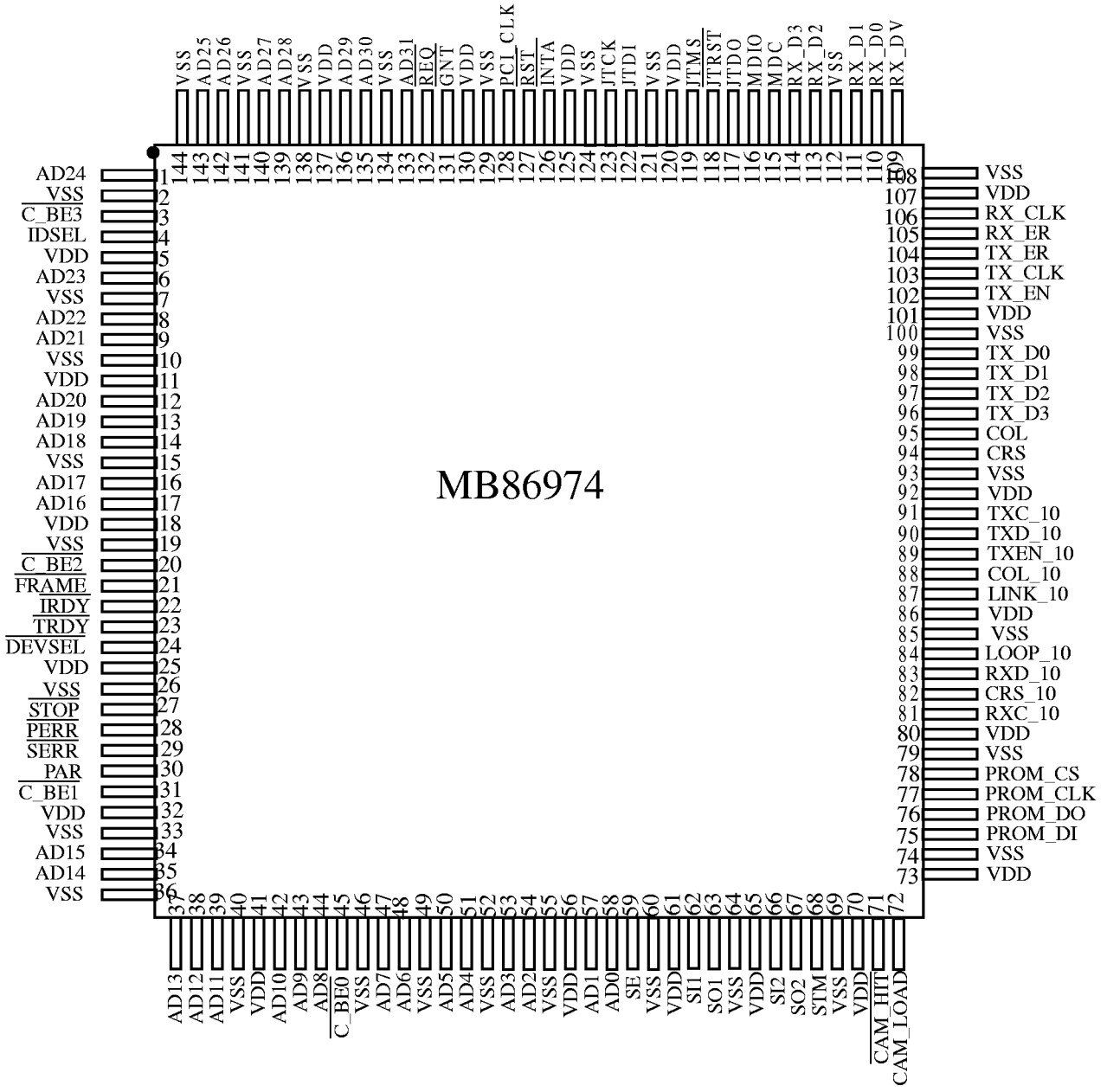
Scan Data Outputs. Output bus carries serial output data shifted from the internal scan chains.

STM

Scan Test Mode. Input asserted during scan testing and deasserted for normal operation. It is required to place the chip in a testable state. This signal should be deasserted when mounted in most user

Package Pinout

The following diagram shows the package pinout for the MB86974:



Electrical Specifications

ABSOLUTE MAXIMUM RATINGS

Parameter	Value	Unit	Notes
Vcc (Measured to Vss)	4.0	V	
Digital Inputs/Outputs	Vss - 0.5 to Vcc + 2.5 or Vss + 5.5	V	
Ambient Operating Temperature	70	°C	
Continuous Power Dissipation	500	mW	
Storage Temperature	-65 to 150	°C	
Junction Temperature	110	°C	
Soldering Temperature	260	°C	
Input Current on any signal pin	+/- 150	mA	1,2
Ouput Current on any PCI Output pin	150	mA	2,3
Ouput Current on any other Output pin	50	mA	2

Note: 1. Limiting current below which Latchup will not occur..
2. Only one pin at a time, for short periods.
3. Excluding $\overline{\text{INTA}}$, which is a standard strength output.

Warning: Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of the specifications is not implied. Exposure to absolute maximum rating conditions for extended periods can affect device reliability.

Electrostatic Discharge

This (and all electronic components) should be protected from accidental electronic discharge. Samples of the MB86974 have been tested to withstand repeated discharges using the JEDEC 'Human Body Model' at 2kV, and the 'Machine Model' at 200V.

Recommended Operating Conditions

Parameter	Symbol	Min	Max	Unit
Ambient Operating Temperature	T_A	0	+70	°C
Vcc Supply Voltage	Vcc	3.0	3.6	V

Notes:-The "Recommended Operating Conditions" are normal operating ranges for the device. Operation outside these ranges may adversely affect reliability.

No warranty is made with respect to uses, operating conditions, or combinations not represented on the data sheet. Users considering application outside these ranges are advised to consult their FUJITSU representative beforehand.

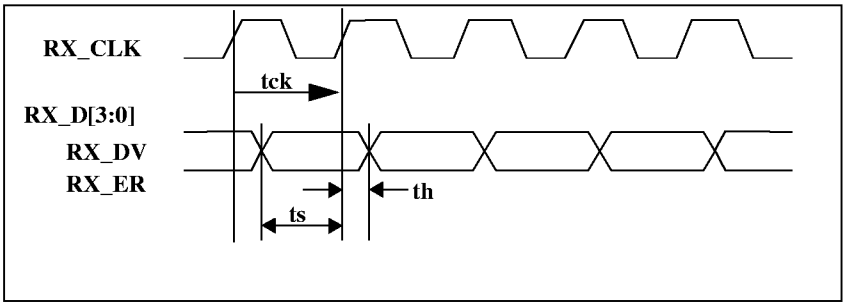
DC Characteristics (Ta = 0-70^0 C, Vcc = 3.3V+/- ±10%)

Parameter	Symbol	Conditions	Min	Typ	Max	Unit	Note
Quiescent Power Supply Current	I_{CCs}			33	60	mA	
Dynamic Power Supply Current	I_{CCd}	PCI_CLK = 33Mhz		80	100	mA	
Power Consumption	pwr			330		mW	
Output High Current (Switching)	Ioh(AC)	Voh = 0.7VccV			32Vcc	mA	1,2
(PCI pins) (by design)	Ioh(AC)	Voh = 0.3VccV	12Vcc			mA	1,2
Output High Voltage (PCI pins)	Voh	Ioh = -.5 mA	0.9Vcc			V	1
Output High Voltage (all other pins)	Voh	Ioh = -4 mA	2.4			V	
Output Low Current (Switching)	Iol(AC)	Vol = 0.18VccV			38Vcc	mA	2,3
(PCI pins) (by design)	Iol(AC)	Voh = 0.6VccV	16Vcc			mA	2,3
Output Low Voltage (PCI pins)	Vol	Ioh = 1.5 mA			0.1Vcc	V	3
Output Low Voltage (all other pins)	Vol	Ioh = 4 mA			0.4	V	
Input High Voltage (PCI pins)	Vih		0.5Vcc		Vcc	V	
Input High Voltage (all other pins)	Vih		2.0			V	
Input Low Voltage (PCI pins)	Vil				0.325Vcc	V	
Input Low Voltage (all other pins)	Vil				0.8	V	
Input Low Current ($\overline{\text{CAM\_HIT}}$, PROM_DI, $\overline{\text{GNT}}$)	Iinl	Vcc-max, Vin=GND	-200	-100	-30	µA	
Input Low Current (all others)	Iinl	Vcc-max, Vin=GND	-10	+/-.05	10	µA	
Input High Current ($\overline{\text{CAM\_HIT}}$, PROM_DI, $\overline{\text{GNT}}$)	Iinh	Vcc-max, Vin=Vcc	-10	-5	10	µA	
Input High Current (all others)	Iinh	Vcc-max, Vin=Vcc	-10	+/-.05	10	µA	
Output Tristate Leakage Current	Ioz		-10	+/-.05	10	µA	
Input Pin Capacitance (IDSEL pin)	C_{IDSEL}			5.6	8	pF	2
Input Pin Capacitance (PCI _CLK)	C_{inPCI}			5.6	12	pF	2
Input Pin Capacitance (PCI pins)	C_{inPCI}			7.9	10	pF	2,4
Input Pin Capacitance (all others)	C_{inPCI}			5.6	8	pF	2

Note: 1. Excluding $\overline{\text{SERR}}$ and $\overline{\text{INTA}}$, which are open-drain outputs.
2. Sample tested only, not tested in production.
3. Excluding $\overline{\text{INTA}}$, which is a standard strength output.
4. Excluding PCI_CLK, $\overline{\text{RST}}$, $\overline{\text{GNT}}$, $\overline{\text{INT\_A}}$, IDSEL

Timing Diagrams

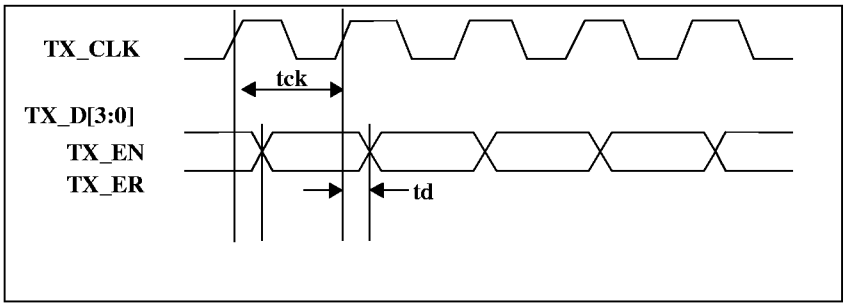
Receive Cycle (MII)



Receive Cycle Timings (MII)

Timing	Parameter	Min	Typ	Max	Unit
ts	RX_D[3:0], RX_DV, RX_ER Setup to RX_CLK Rising Edge	10	3		nS
th	RX_D[3:0], RX_DV, RX_ER Hold after RX_CLK Rising Edge	12			nS
tck	RX_CLK period	33	40		nS

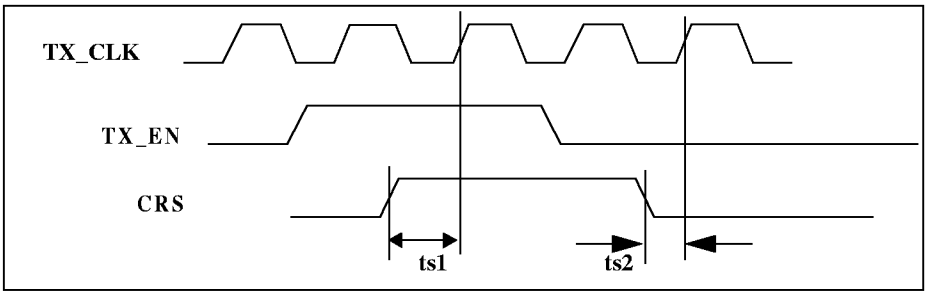
Transmit Cycle (MII)



Transmit Cycle Timings (MII)

Timing	Parameter	Min	Typ	Max	Unit
td	TX_D[3:0], TX_EN, TX_ER Delay from TX_CLK Rising Edge	0	10	13	nS
tck	TX_CLK period	33	40		nS

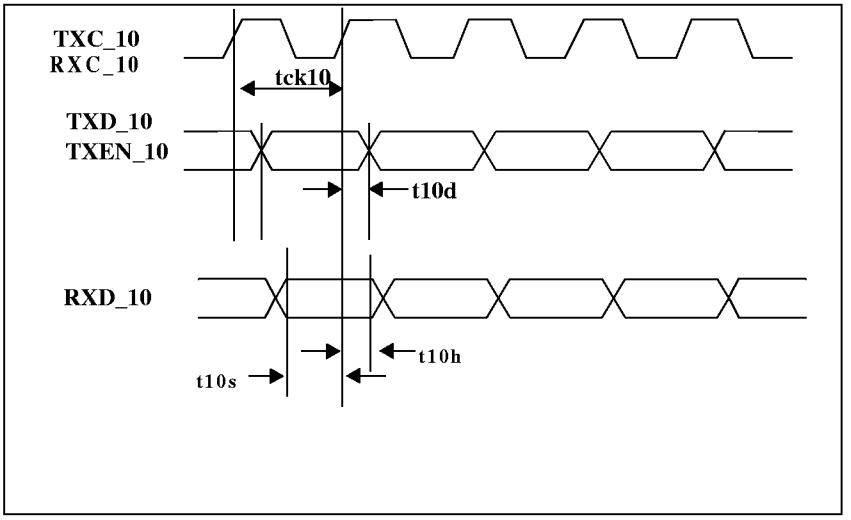
Carrier Sense Assertion on a Transmit Cycle



Carrier Sense Timing

Timing	Parameter	Min	Typ	Max	Unit
ts1	CRS assertion setup to TX_CLK		3	8	nS
ts2	CRS de-assertion setup to TX_CLK		3	8	nS

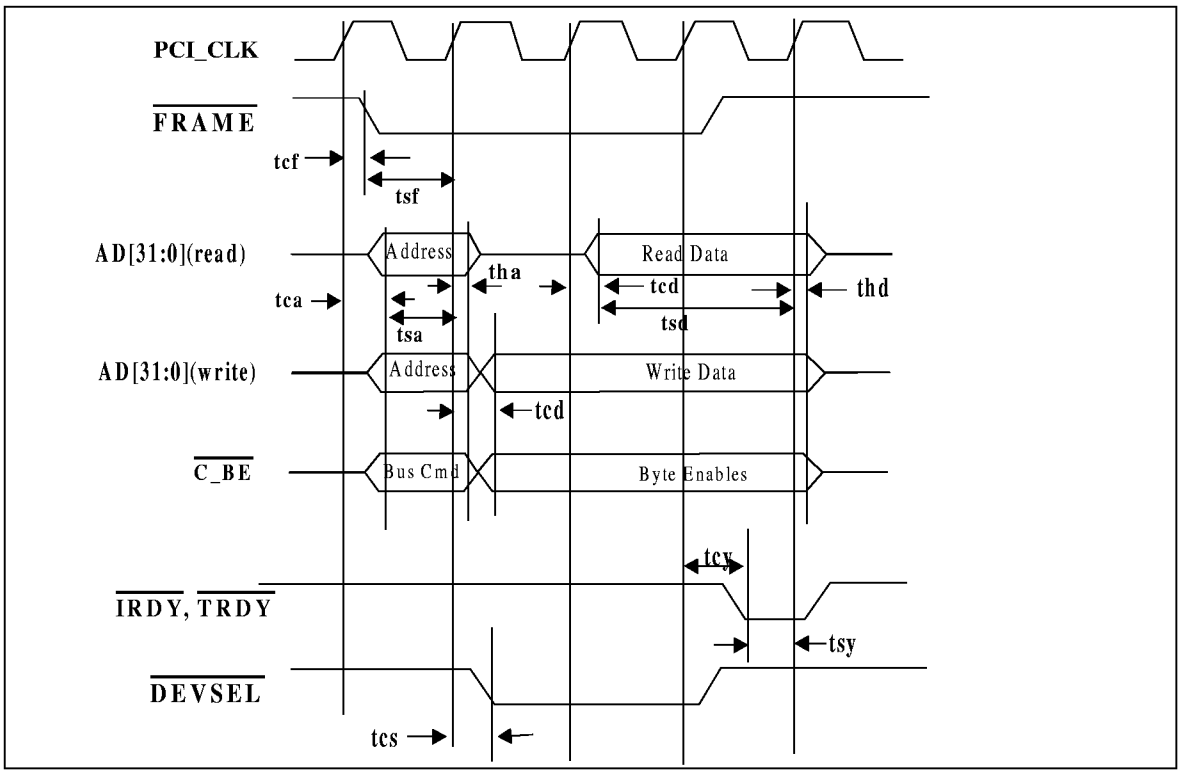
10Mb/s Interface (7-wire) Timing



10Mb/s Interface Timing

Timing	Parameter	Min	Typ	Max	Unit
t10d	TXD_10, TXEN_10 Delay from TXC_10	0	10	13	nS
tck10	TXC_10, RXC_10 period	33	100		nS
t10s	RXD_10 Setup to RXC_10		3	10	
t10h	RXD_10 Hold to RXC_10			10	

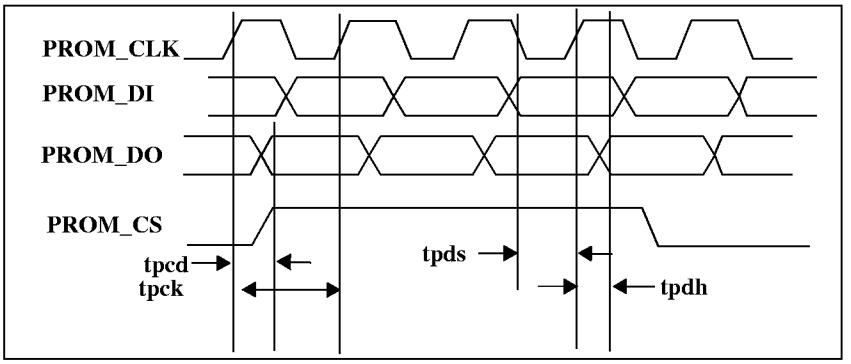
PCI Cycle Timing Diagram



PCI Cycle Timing

Timing	Init/Trgt	Parameter	Min	Typ	Max	Unit
tcf	Init	FRAME Delay from PCI_CLK Rising Edge	2	8.5	13	nS
tsf	Target	FRAME Setup to PCI_CLK Rising Edge	7	5		nS
tca	Init	AD[31:0], C_BE[3:0] delay from PCI_CLK	2	8.5	13	nS
tsa	Target	AD[31:0], C_BE[3:0] Setup to PCI_CLK	7	5		nS
tha	Target	AD[31:0], C_BE[3:0] Hold after PCI_CLK	4.5	2.5		nS
tcd	Source	AD[31:0] Delay from PCI_CLK	2	8.5	13	nS
tsd	Destin	AD[31:0] Setup to PCI_CLK Rising Edge	7	5		nS
thd	Destin	AD[31:0] Hold after PCI_CLK Rising Edge	4.5	2.5	0	nS
tcy	One each	IRDY, TRDY Delay from PCI_CLK	2	8.5	13	nS
tsy	One each	IRDY, TRDY Setup to PCI_CLK	7	5		nS
tcs	Target	DEVSEL delay from PCI_CLK	2	8.5	13	nS

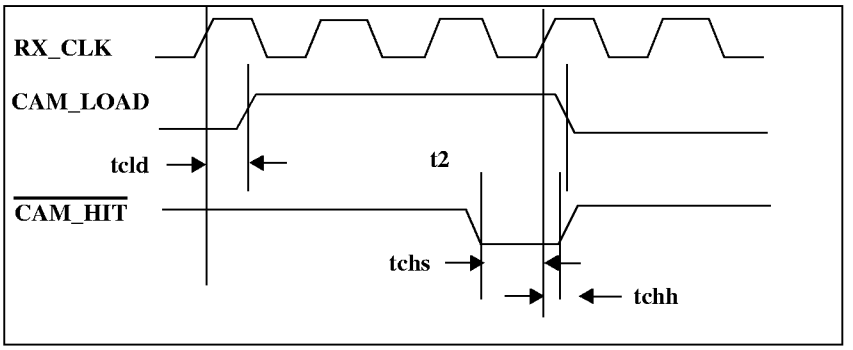
EEPROM Timing



EEPROM Timing

Timing	Parameter	Min	Typ	Max	Unit
tpck	PROM_CK period	1000	1030		nS
tpcd	PROM_DO, PROM_CS delay from PROM_CK		11	15	nS
tpds	PROM_DI Setup to PROM_CK	7	5		
tpdh	PROM_DI Hold from PROM_CK	5	3		

External CAM Timing



External CAM Timing

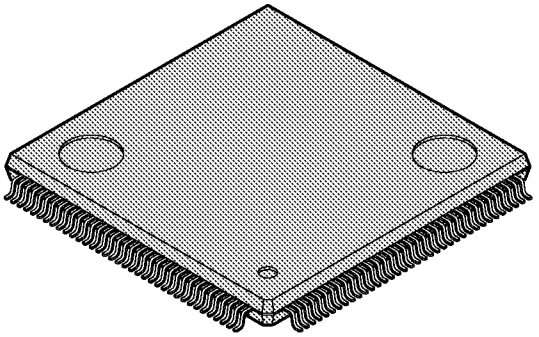
Timing	Parameter	Min	Typ	Max	Unit
tcld	CAM_LOAD Delay from RX_CLK			20	nS
tchs	$\overline{\text{CAM\_HIT}}$ Setup to RX_CLK	10			
tchh	$\overline{\text{CAM\_HIT}}$ Hold from RX_CLK	8			nS

ORDERING INFORMATION

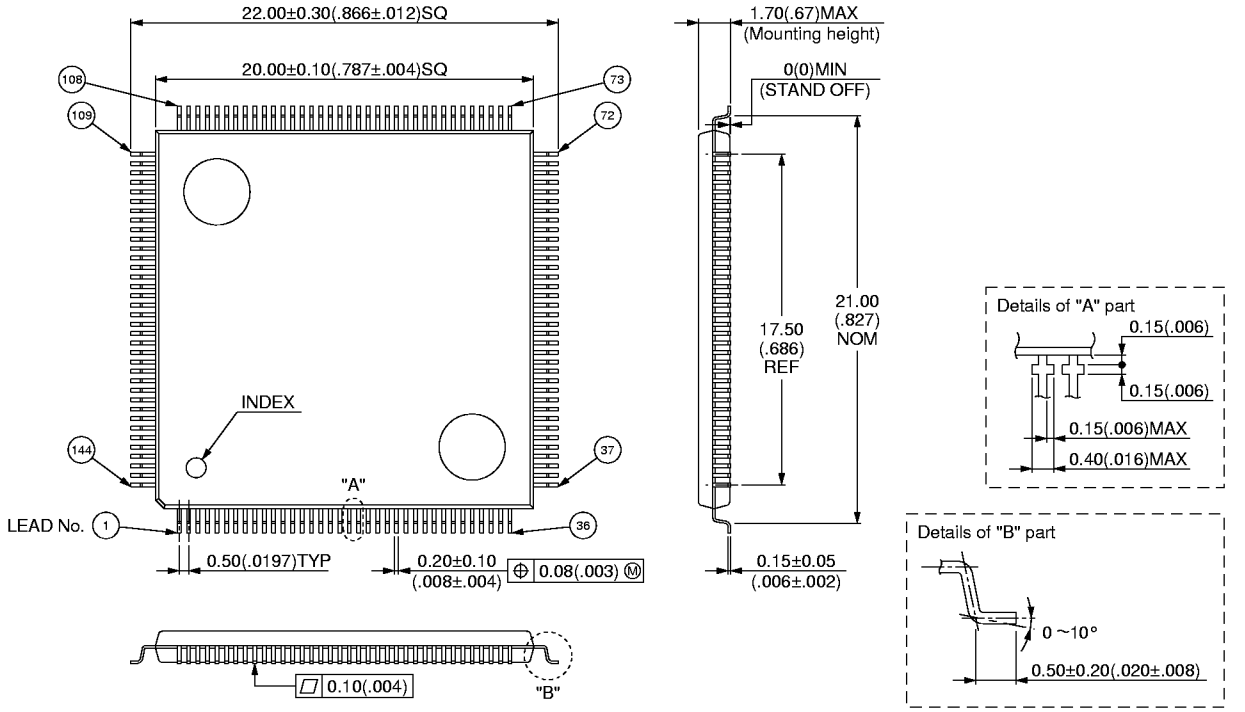
Package	Order Number
144-pin PQFP	MB86974APFV

PACKAGE DRAWING

FPT-144P-M08

144-pin plastic LQFP		
 (FPT-144P-M08)	Lead pitch	0.50 mm
	Package width × package length	20 × 20 mm
	Lead shape	Gullwing
	Sealing method	Plastic mold
	Mounting height	1.70 mm MAX

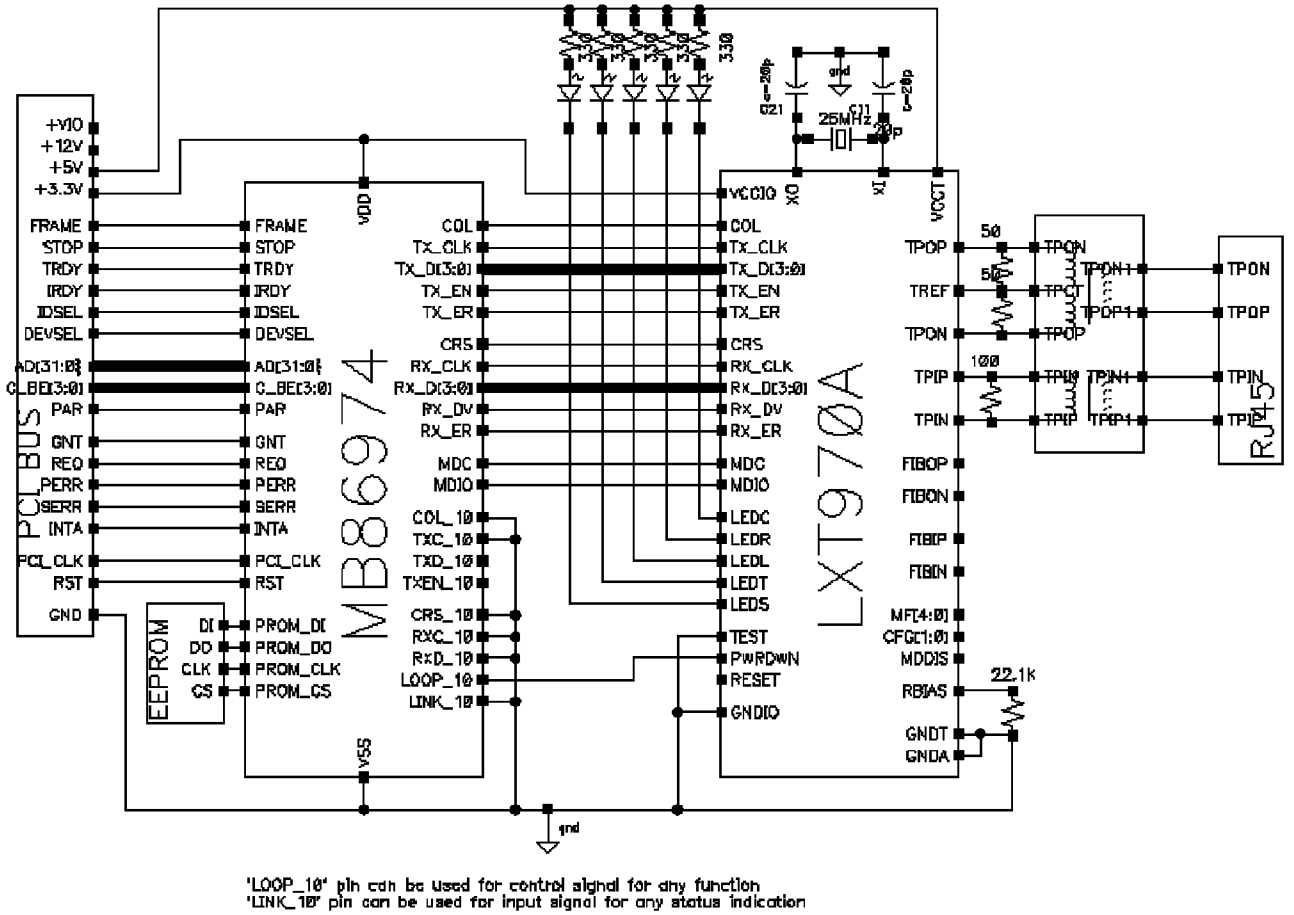
144-pin plastic LQFP
(FPT-144P-M08)



Dimensions in mm (inches).

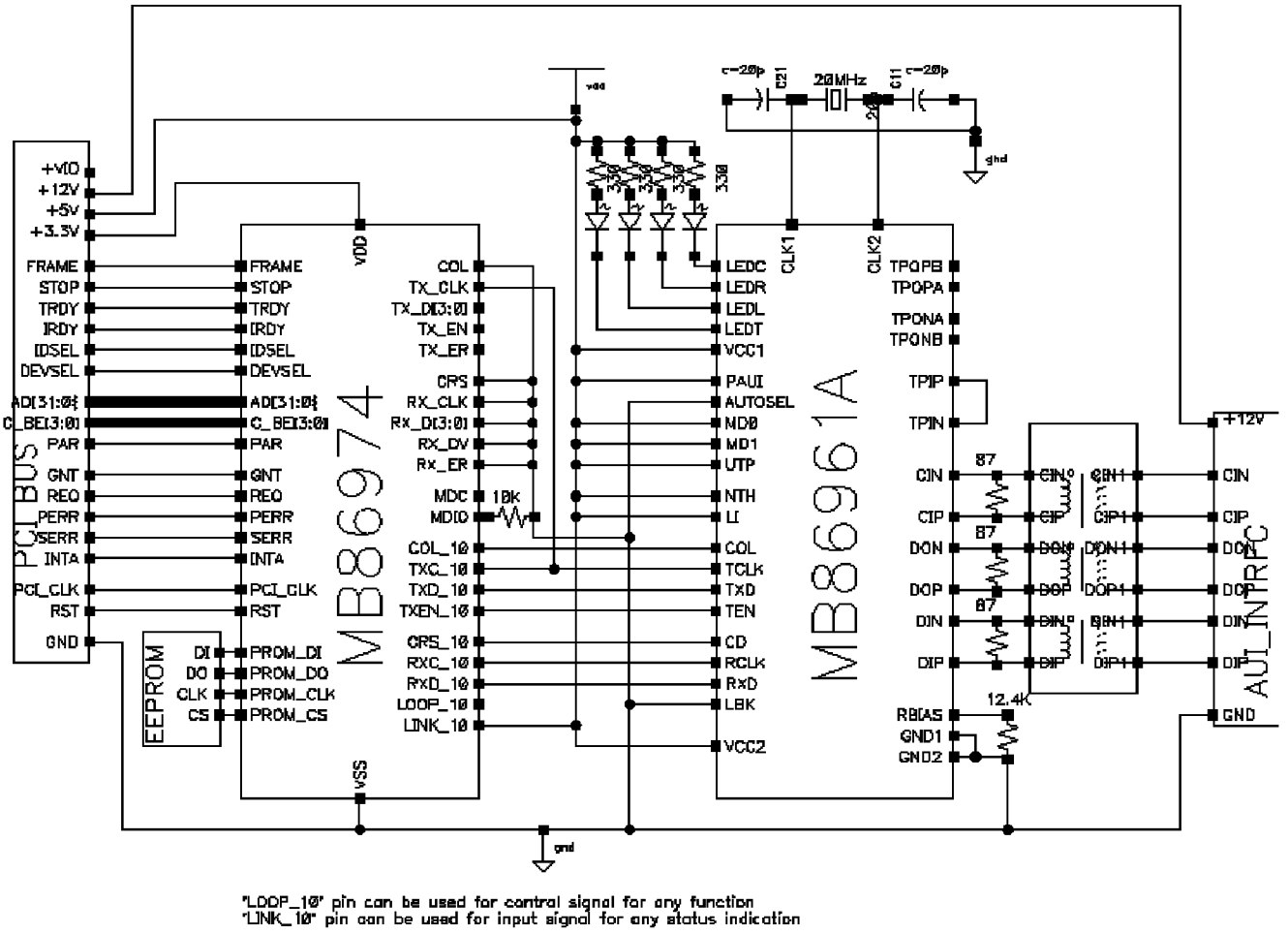
Typical Applications

Typical NIC Card Application



This schematic shows the MB86974 interfacing between a PCI bus and a typical 10/100 MII-based Phy. The configuration options for the Phy are not shown, since they depend on the actual configuration desired. No external CAM is provided, but the Phy does include support for 100Base-FX interface. The LOOP_10 output from the MB86974 is used to drive the PWRDWN pin on the Phy, so that software can be used to power down the Ethernet system, for instance in the absence of a network connection. The LINK_10 input could be used to indicate to software the presence of an add-in board (e.g. a 100Base-FX board), or for some other status indication.

PCI to AUI Interface

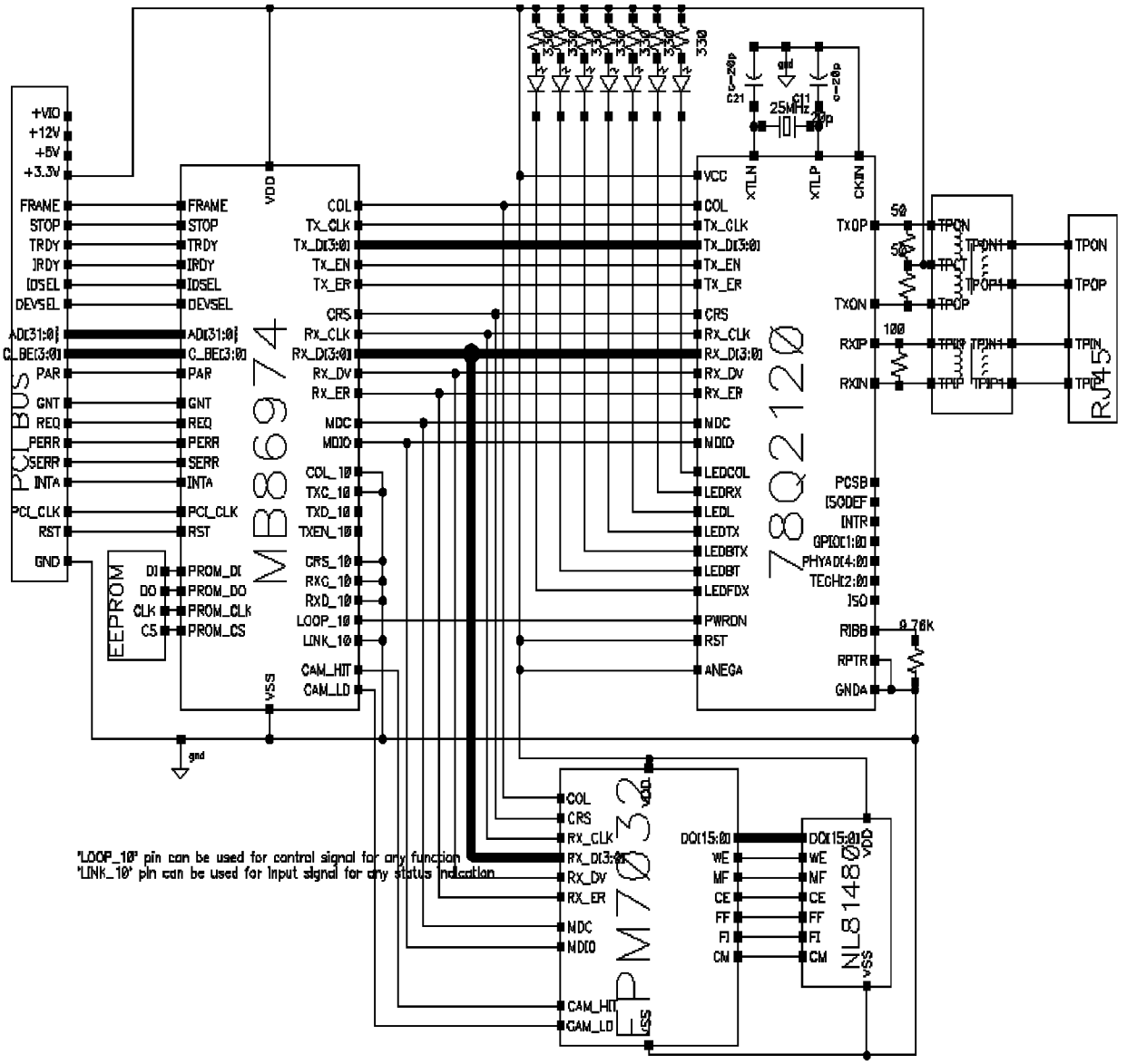


"LOOP_10" pin can be used for control signal for any function
"LINK_10" pin can be used for input signal for any status indication

This shows the MB86974 with an AUI interface, using the MB86961A device, via the 10Mb endec ('7-wire' interface. The 10Base-T connections of the MB86961A are not used, and the MB86974 MII inputs are disabled. A clock is provided to the MII TX_CLK input to ensure that the internal state machines correctly process the MII-'7-wire' transition. For a 10Base-2 or 10Base-5 system, the LOOP_10 pin can be used to control an isolated power supply for the actual transceiver.

This circuit can be combined with the previous circuit to provide an adapter that will handle 10Base-5, 10Base-2, 10Base-T, 100Base-TX and 100Base-FX connections. The software driver would wait for auto-negotiation to end. If a 100Base-TX or 10Base-T response was found, the system would use that. If neither of these was found, then the 100Base-FX can be tested. If Link is established, then the system uses that, but if not, the 10Base-2/5 interface can be powered up (using LOOP_10 as the control) and that interface used. Since the 10Base2/5 system has no 'link detect' function, it would be advisable to keep track of the reception of any packet over this interface, and if none appears, then reset the system to the initial configuration and restart autonegotiation. An alternative would be to check for excessive collisions, since a 10Base-2/5 transmit packet will always generate collisions if it is not connected to a properly-terminated cable.

External CAM Interface



This shows another 10/100Mb/s Phy attached to the MB86974, with an external CAM memory. The FPGA 'snoops' the RXD[3:0] and RXCLK lines, and also CRS, COL, RD_DV and RX_ER lines, together with the CAM_LOAD and $\overline{\text{CAM\_HIT}}$ lines. From these, it is able to extract the destination addresses of incoming packets, and accept or reject them as appropriate. The MDC and MDIO lines are available to set up the appropriate addresses.