



STANDARD  
MICROSYSTEMS  
CORPORATION

**LAN91C100**  
**ADVANCE INFORMATION**

## **FEAST™**

### **Fast Ethernet Controller**

#### **FEATURES**

- Dual Speed CSMA/CD Engine (10 Mbps and 100 Mbps)
- Compliant with IEEE 802.3 100BASE-T Specification
- Supports 100BASE-TX, 100BASE-T4, and 10BASE-T Physical Interfaces
- 32 Bit Wide Data Path (Into Packet Buffer Memory)
- Support for 32 and 16 Bit Buses
- Support for 32, 16 and 8 Bit CPU Accesses
- Synchronous, Asynchronous and Burst DMA Interface Mode Options
- 128 Kbyte External Memory
- Built-in Transparent Arbitration for Slave Sequential Access Architecture
- Flat MMU Architecture with Symmetric Transmit and Receive Structures and Queues
- MII (Media Independent Interface) Compliant MAC-PHY Interface (Compliant with Emerging MII Standard Interface)
- MII Management Serial Interface
- Seven Wire Interface to 10 Mbps ENDEC (LAN83C694)
- EEPROM-Based Setup
- 208 Pin PQFP and TQFP Package

#### **GENERAL DESCRIPTION**

The LAN91C100 FEAST is a high-speed network controller designed to facilitate the implementation of Fast Ethernet adapters and connectivity products. It contains a dual speed CSMA/CD engine that implements the MAC portion of the CSMA/CD protocol at 10 and 100 Mbps and couples it with a lean and fast data and control path system architecture to ensure data movement with no bottlenecks at 100 Mbps.

Memory management is handled using a unique MMU (Memory Management Unit) architecture and a 32-bit wide data path. This I/O mapped architecture can sustain back-to-back frame transmission and reception for superior data throughput and optimal performance. It also

dynamically allocates buffer memory in an efficient buffer utilization scheme, reducing software tasks and relieving the host CPU from performing these housekeeping functions. The total memory size is 128 Kbytes (external), equivalent to a total chip storage (transmit and receive) of 64 outstanding packets.

FEAST provides a flexible slave interface for easy connectivity with industry-standard buses. The Bus Interface Unit (BIU) can handle synchronous as well as asynchronous buses, with different signals being used for each one. FEAST's bus interface supports synchronous buses like the VESA local bus, as well as burst mode DMA for EISA environments. Asynchronous bus support for ISA is supported

## TABLE OF CONTENTS

FEATURES .....	1
GENERAL DESCRIPTION.....	1
PIN CONFIGURATION.....	3
DESCRIPTION OF PIN FUNCTIONS .....	4
FUNCTIONAL DESCRIPTION .....	14
DATA STRUCTURES AND REGISTERS .....	17
BOARD SETUP INFORMATION.....	59
APPLICATION CONSIDERATIONS.....	62
OPERATIONAL DESCRIPTION.....	69
MAXIMUM GUARANTEED RATINGS .....	69
DC ELECTRICAL CHARACTERISTICS .....	69
TIMING DIAGRAMS.....	72



80 Arkay Drive  
Hauppauge, NY 11788  
(516) 435-6000  
FAX (516) 273-3123

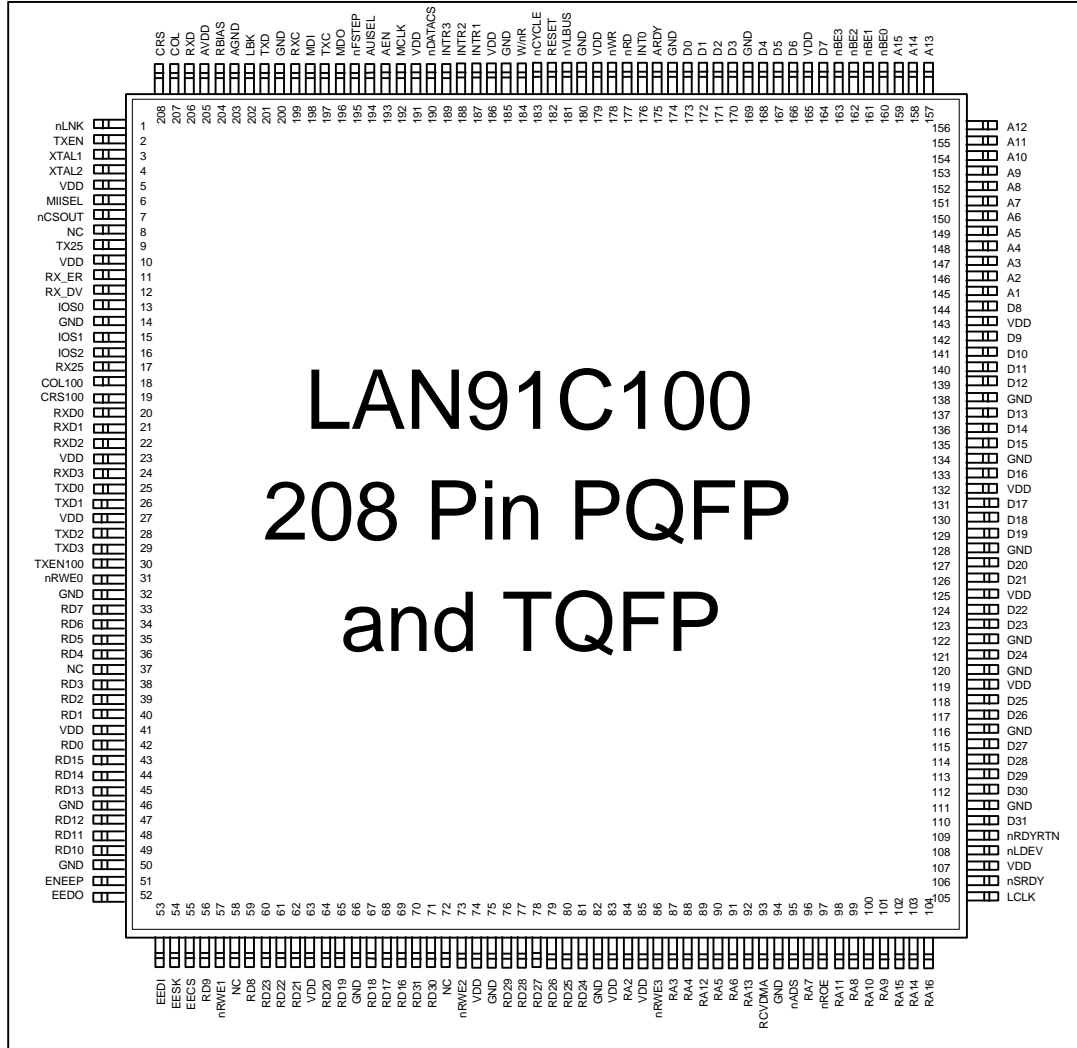
even though ISA cannot sustain 100 Mbps traffic. Fast Ethernet could be adopted for ISA-based nodes on the basis of the aggregate traffic benefits.

FEAST is software-compatible with the existing LAN9000 family of products and can use current LAN9000 drivers in 16- and 32-bit Intel X86-based environments.

Two different interfaces are supported on the network side. The first is a conventional seven

wire ENDEC interface that connects to the LAN83C694 for 10BASE-T and coax 10 Mbps Ethernet networks. The second interface follows the MII (Media Independent Interface) specification draft standard, consisting of 4 bit wide data transfers at the nibble rate. FEAST also interfaces to the MII serial management protocol. Four I/O ports (one input and three output pins) are provided for LAN83C694 configuration.

### PIN CONFIGURATION



**DESCRIPTION OF PIN FUNCTIONS OF PIN FUNCTIONS PIN FUNCTIONS**

<b>PQFP/TQFP PIN NO.</b>	<b>NAME</b>	<b>SYMBOL</b>	<b>BUFFER TYPE</b>	<b>DESCRIPTION</b>
148-159	Address	A4-A15	I	Input. Decoded by the LAN91C100 to determine accesses to its registers.
145-147	Address	A1-A3	I	Input. Used by the LAN91C100 for internal register selection.
193	Address Enable	AEN	I	Input. Used as an address qualifier. Address decoding is only enabled when AEN is low.
160-163	nByte Enable	nBE0-nBE3	I	Input. Used during LAN91C100 register accesses to determine the width of the access and the register(s) being accessed. nBE0-nBE3 are ignored when nDATACS is low (burst accesses) because 32 bit transfers are assumed.
173-170, 168-166, 164,144, 142-139, 137-135, 133, 131-129, 127,126, 124,123, 121,118, 117, 115-112, 110	Data Bus	D0-D31	I/O24	Bidirectional. 32 bit data bus used to access the LAN91C100's internal registers. Data bus has weak internal pullups. Supports direct connection to the system bus without external buffering. For 16 bit systems, only D0-D15 are used.
182	Reset	RESET	IS	Input. This input is not considered active unless it is active for at least 100ns to filter narrow glitches.
95	nAddress Strobe	nADS	IS	Input. Address strobe. For systems that require address latching, the rising edge of nADS indicates the latching moment for A1-A15 and AEN. All LAN91C100 internal functions of A1-A15, AEN are latched except for nLDEV decoding.
183	nCycle	nCYCLE	I	Input. This active low signal is used to control LAN91C100 synchronous bus cycles.

**DESCRIPTION OF PIN FUNCTIONSOFPIN FUNCTIONSPIN FUNCTIONS**

<b>PQFP/TQFP PIN NO.</b>	<b>NAME</b>	<b>SYMBOL</b>	<b>BUFFER TYPE</b>	<b>DESCRIPTION</b>
184	Write/nRead	W/nR	I	Input. Defines the direction of synchronous cycles. Write cycles when high, read cycles when low.
181	nVL Bus Access	nVLBUS	IP	Input. When low the LAN91C100 synchronous bus interface is configured for VL Bus accesses. Otherwise the LAN91C100 is configured for EISA DMA burst accesses. Does not affect the asynchronous bus interface.
105	Local Bus Clock	LCLK	I	Input. Used to interface synchronous buses. Maximum frequency is 50 MHz. Limited to 8.33 MHz for EISA DMA burst mode.
175	Asynchronous Ready	ARDY	OD16	Open drain output. ARDY may be used when interfacing asynchronous buses to extend accesses. Its rising (access completion) edge is controlled by the XTAL1 clock and therefore asynchronous to the host CPU or bus clock.
106	nSynchronous Ready	nSRDY	O16	Output. This output is used when interfacing synchronous buses and nVLBUS=0 to extend accesses. This signal remains normally inactive, and its falling edge indicates completion. This signal is synchronous to the bus clock LCLK.
109	nReady Return	nRDYRTN	I	Input. This input is used to complete synchronous read cycles. In EISA burst mode it is sampled on falling LCLK edges, and synchronous cycles are delayed until it is sampled high.
176 187-189	Interrupt	INT0-INT3	O24	Outputs. Only one of these interrupts is selected to be used; the other three are tri-stated. The selection is determined by the value of INT SEL1-0 bits in the Configuration Register.

**DESCRIPTION OF PIN FUNCTIONSOF PIN FUNCTIONSPIN FUNCTIONS**

<b>PQFP/TQFP PIN NO.</b>	<b>NAME</b>	<b>SYMBOL</b>	<b>BUFFER TYPE</b>	<b>DESCRIPTION</b>
108	nLocal Device	nLDEV	O16	Output. This active low output is asserted when AEN is low and A4-A15 decode to the LAN91C100 address programmed into the high byte of the Base Address Register. nLDEV is a combinatorial decode of unlatched address and AEN signals.
177	nRead Strobe	nRD	IS	Input. Used in asynchronous bus interfaces.
178	nWrite Strobe	nWR	IS	Input. Used in asynchronous bus interfaces.
190	nData Path Chip Select	nDATACS	IP	Input. When nDATACS is low, the Data Path can be accessed regardless of the values of AEN, A1-A15 and the content of the BANK SELECT Register. nDATACS provides an interface for bursting to and from the LAN91C100 32 bits at a time.
54	EEPROM Clock	EESK	O4	Output. 4 $\mu$ sec clock used to shift data in and out of the serial EEPROM.
55	EEPROM Select	EECS	O4	Output. Used for selection and command framing of the serial EEPROM.
52	EEPROM Data Out	EEDO	O4	Output. Connected to the DI input of the serial EEPROM.
53	EEPROM Data In	EEDI	ID	Input. Connected to the DO output of the serial EEPROM.
13,15,16	I/O Base	IOS0-IOS2	IP	Input. External switches can be connected to these lines to select between predefined EEPROM configurations.
51	Enable EEPROM	ENEPP	IP	Input. Enables (when high or open) LAN91C100 accesses to the serial EEPROM. Must be grounded if no EEPROM is connected to the LAN91C100.

**DESCRIPTION OF PIN FUNCTIONSOFPIN FUNCTIONSPIN FUNCTIONS**

<b>PQFP/TQFP PIN NO.</b>	<b>NAME</b>	<b>SYMBOL</b>	<b>BUFFER TYPE</b>	<b>DESCRIPTION</b>
42, 40-38, 36-33, 59,56, 49-47, 45-43, 69-67, 65,64, 62-60, 81-76, 71,70	RAM Data Bus	RD0-RD31	I/O4P	Bidirectional. Carries the local buffer memory read and write data. Reads are always 32 bits wide. Writes are controlled individually at the byte level.
84,87, 88,90, 91,96, 99,101, 100,98, 89,92, 103,102, 104	RAM Address Bus	RA2-RA16	O4	Outputs. This bus specifies the buffer RAM doubleword being accessed by the LAN91C100.
97		nROE	O4	Output. Used to read a doubleword from buffer RAM.
31,57, 73,86		nRWE0- RWE3	O4	Outputs. Used to write any byte, word or dword in RAM.
93	nReceive DMA	nRCVDMA	O4	Output. This pin is active during LAN91C100 write memory cycles of receive packets.
3 4	Crystal 1 Crystal 2	XTAL1 XTAL2	I <sub>CLK</sub>	An external 25 MHz crystal is connected across these pins. If a TTL clock is supplied instead, it should be connected to XTAL1 and XTAL2 should be left open.
5,10, 23,27, 41,63, 74,83, 85,107, 119,125, 132,143, 165,179, 186,191	Power	VDD		+5V power supply pins.
205	Analog Power	AVDD		+5V analog power supply pin.

**DESCRIPTION OF PIN FUNCTIONS OF PIN FUNCTIONS PIN FUNCTIONS**

<b>PQFP/TQFP PIN NO.</b>	<b>NAME</b>	<b>SYMBOL</b>	<b>BUFFER TYPE</b>	<b>DESCRIPTION</b>
14,32, 46,50, 66,75, 82,94, 111,116, 120,122, 128,134, 138,169, 174,180, 185,200	Ground	GND		Ground pins.
203	Analog Ground	AGND		Analog ground pin.
2	Transmit Enable	TXEN	O4	Output. Used for 10 Mbps ENDEC. This pin stays low when MIISEL is high.
201	Transmit Data	TXD	O4	NRZ transmit data output for 10 Mbps ENDEC interface.
208	Carrier Sense	CRS	ID	Input. Carrier sense from 10 Mbps ENDEC interface. This pin is ignored when MIISEL is high.
207	Collision Detection	COL	ID	Input. Collision detection indication from 10 Mbps ENDEC interface. This pin is ignored when MIISEL is high.
206	Receive Data	RXD	IP	NRZ receive data input from 10 Mbps ENDEC interface. This pin is ignored when MIISEL is high.
197	Transmit Clock	TXC	IP	Input. 10 MHz transmit clock used in 10 Mbps operation. This pin is ignored when MIISEL is high.
199	Receive Clock	RXC	IP	Input. 10Mhz receive clock recovered by the 10 Mbps ENDEC. This pin is ignored when MIISEL is high.
202	Loopback	LBK	O4	Output. Active when LOOP bit is set (TCR bit 1). Independent of port selection (MIISEL=X).



**DESCRIPTION OF PIN FUNCTIONS OF PIN FUNCTIONS PIN FUNCTIONS**

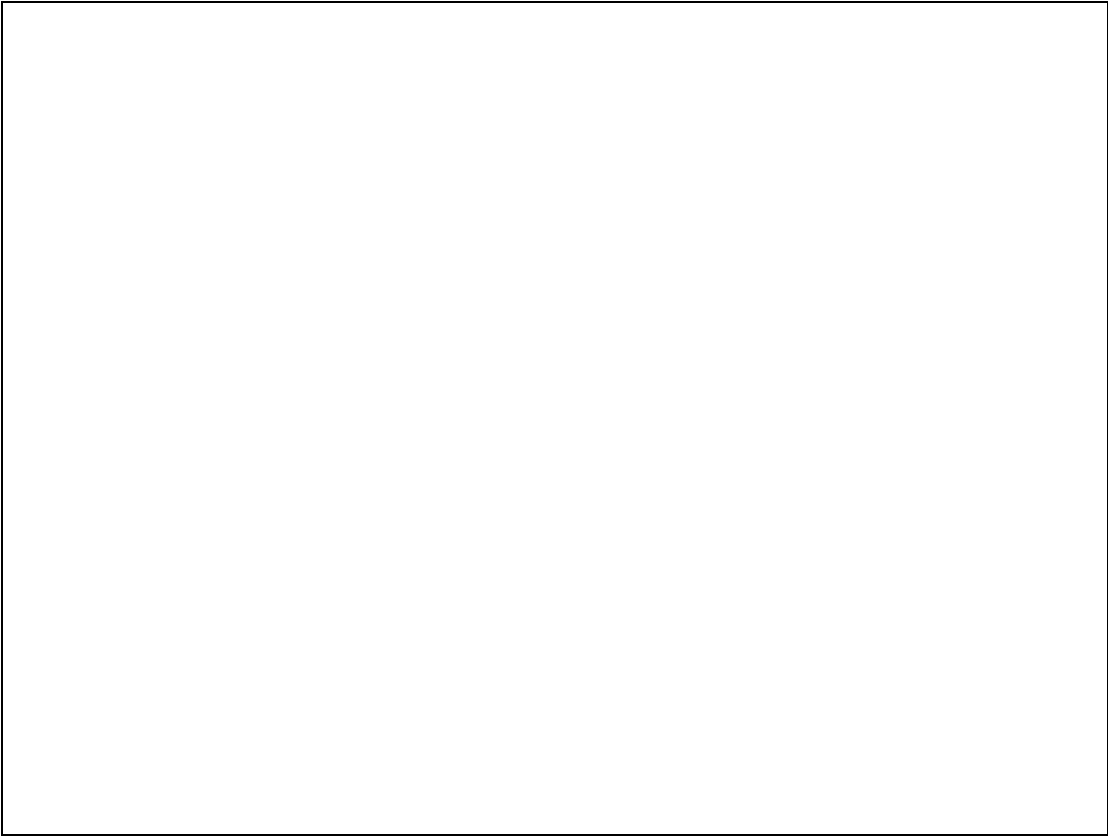
<b>PQFP/TQFP PIN NO.</b>	<b>NAME</b>	<b>SYMBOL</b>	<b>BUFFER TYPE</b>	<b>DESCRIPTION</b>
1	nLink Status	nLNK	IP	Input. General purpose input port used to convey LINK status (EPHSR bit 14). Independent of port selection (MIISEL=X).
195	nFullstep	nFSTEP	O4	Output. Non volatile output pin. Driven by inverse of FULLSTEP (CONFIG bit 10). Independent of port selection (MII SEL=X).
6	MII Select	MIISEL	O4	Output. Non volatile output pin. Driven by MII SELECT (CONFIG bit 15). High indicates the MII port is selected, low indicates the 10 Mbps ENDEC is selected.
194	AUI Select	AUISEL	O4	Output. Non volatile output pin. Driven by AUI SELECT (CONFIG bit 8). Independent of port selection (MIISEL=X).
30	Transmit Enable 100 Mbps	TXEN100	O4	Output to MII PHY. Envelope to 100 Mbps transmission. This pin stays low if MIISEL is low.
19	Carrier 100 Mbps	CRS100	IP	Input from MII PHY. Envelope of packet reception used for deferral and backoff purposes. This pin is ignored when MIISEL is low.
12	Receive Data Valid	RX_DV	ID	Input from MII PHY. Envelope of data valid reception. Used for receive data framing. This pin is ignored when MIISEL is low.
18	Collision Detect 100 Mbps	COL100	ID	Input from MII PHY. Collision detection input. This pin is ignored when MIISEL is low.
25,26, 28,29	Transmit Data	TXD0-TXD3	O4	Outputs. Transmit Data nibble to MII PHY.
9	Transmit Clock	TX25	IP	Input. Transmit clock input from MII. Nibble rate clock (25 MHz). This pin is ignored when MIISEL is low.
17	Receive Clock	RX25	IP	Input. Receive clock input from MII PHY. Nibble rate clock. This pin is ignored when MIISEL is low.
20,21, 22,24	Receive Data	RXD0-RXD3	I	Inputs. Received Data nibble from MII PHY. These pins are ignored when MIISEL is low.

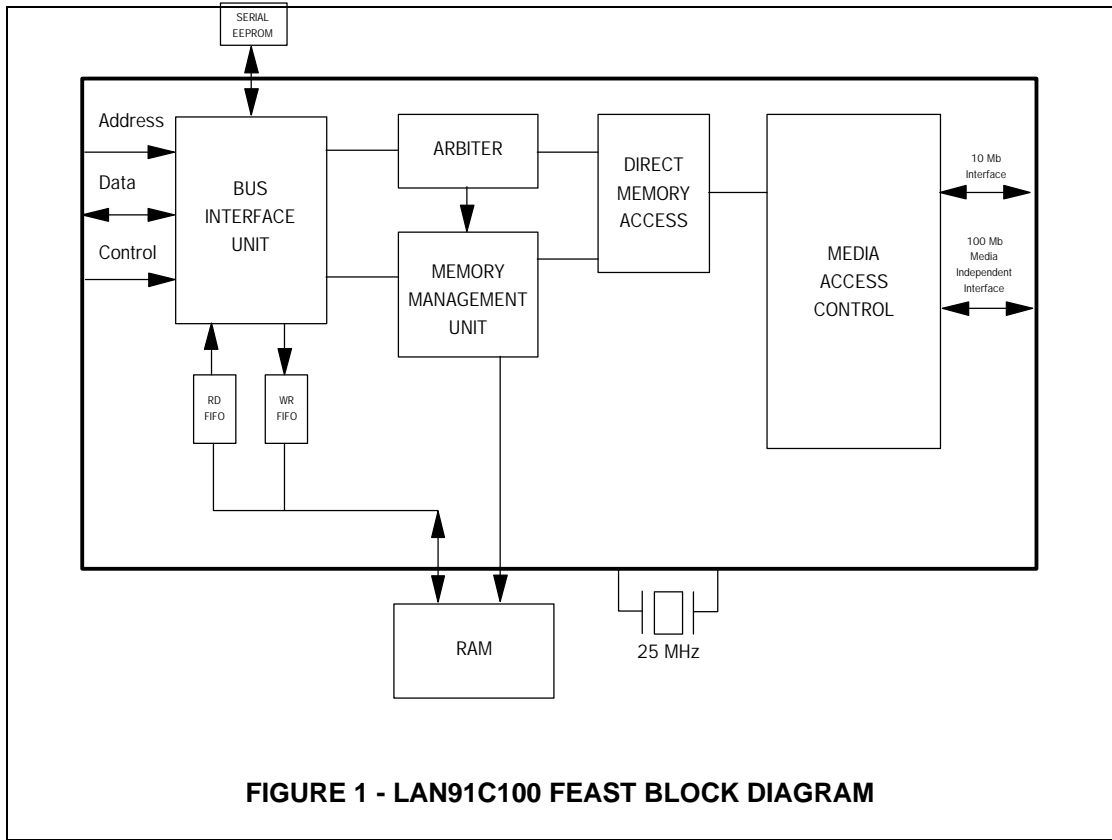
**DESCRIPTION OF PIN FUNCTIONSOFPIN FUNCTIONSPIN FUNCTIONS**

<b>PQFP/TQFP PIN NO.</b>	<b>NAME</b>	<b>SYMBOL</b>	<b>BUFFER TYPE</b>	<b>DESCRIPTION</b>
198	Management Data Input	MDI	IP	MII management data input.
196	Management Data Output	MDO	O4	MII management data output.
192	Management Clock	MCLK	O4	MII management clock.
11	Receive Error	RX_ER	ID	Input. Indicates a code error detected by PHY. Used by the LAN91C100 to discard the packet being received. The error indication reported for this event is the same as a bad CRC (Receive Status Word bit 13). This pin is ignored when MIISEL is low.
204	Bias Resistor	RBIAS	Analog Input	A bias resistor is connected between this pin and ground. <b>Nominal value is TBD.</b>
7	nChip Select Output	nCSOUT	O4	Output. Chip Select provided for mapping of PHY functions into LAN91C100 decoded space. Active on accesses to LAN91C100's eight lower addresses when the BANK SELECTED is 7.

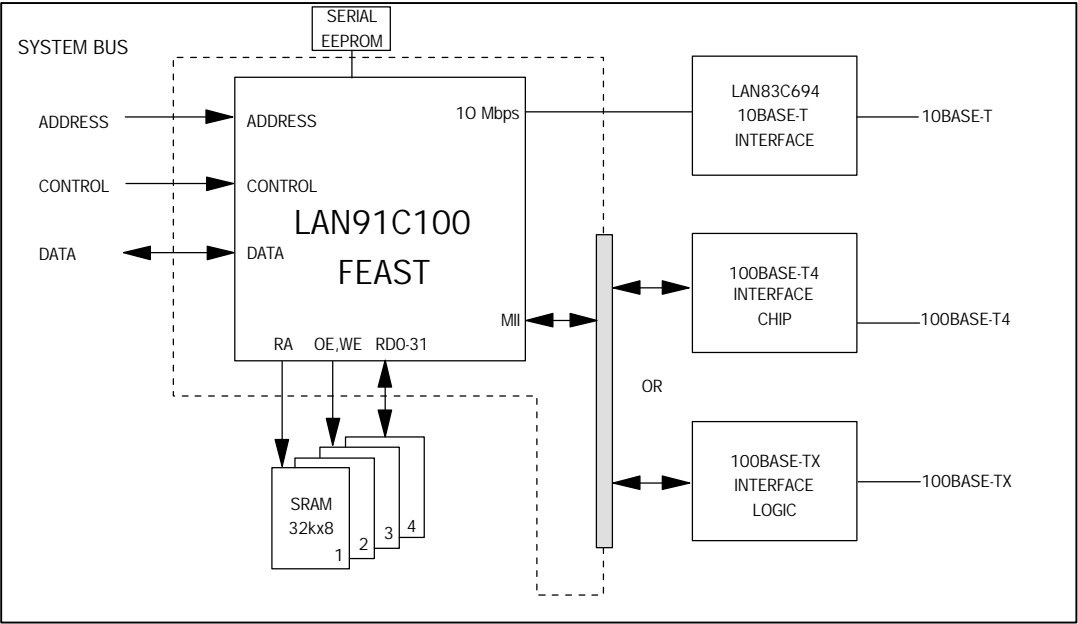
**Table 1 - LAN91C100 Pin Requirements**

<b>FUNCTION</b>	<b>PIN SYMBOLS</b>	<b>NUMBER OF PINS</b>
System Address Bus	A1-A15, AEN, nBE0-nBE3	20
System Data Bus	D0-D31	32
System Control Bus	RESET, nADS, LCLK, ARDY, nRDYRTN, nSRDY, INT0-INT3, nLDEV, nRD, nWR, nDATACS, nCYCLE, W/nR, nVLBUS	17
Serial EEPROM	EEDI, EEDO, EECS, EESK, ENEEP, IOS0-IOS2	8
RAM Data Bus	RD0-RD31	32
RAM Address Bus	RA2-RA16	15
RAM Control Bus	nROE, nRWE0-nRWE3, RCVDMA	6
Crystal Oscillator	XTAL1, XTAL2	2
Power	VDD, AVDD	19
Ground	GND, AGND	21
External ENDEC 10 Mbps	TXEN, TXD, CRS, COL, RXD, TXC, RXC, LBK, nLNK, nFSTEP, AUISEL, MIISEL	12
Physical Interface 100 Mbps	TXEN100, CRS100, COL100, RX_DV, RX_ER, TXD0-TXD3, RXD0-RXD3, MDI, MDO, MCLK	16
Clocks	TX25, RX25	2
Miscellaneous	RBIAS, nCSOUT	2
<b>TOTAL</b>		<b>204</b>





**FIGURE 1 - LAN91C100 FEAST BLOCK DIAGRAM**



**FIGURE 2 - LAN91C100 FEAST SYSTEM DIAGRAM**

## FUNCTIONAL DESCRIPTION

### DESCRIPTION OF BLOCKS

#### Clock Generator Block

The LAN91C100's clock generator uses a 25 MHz crystal connected to pins XTAL1 XTAL2 and generates two free running clocks:

- 1) 50 MHz free running clock - Supplied to the DMA and the ARBITER blocks.
- 2) 25 MHz free running clock - Used to run the EPH during reset or when no TX25 is present.

Other clocks:

- 3) TXCLK and RXCLK are 10 MHz clock inputs. These clocks are generated by the external ENDEC in 10 Mbps mode and are only used by the CSMA/CD block.
- 4) TX25 is an input clock. It will be the nibble rate of the particular PHY connected to the MII (2.5 MHz for a 10 Mbps PHY, and 25 MHz for a 100 Mbps PHY).
- 5) RX25 - This is the MII nibble rate receive clock used for sampling received data nibbles and running the receive state machine (2.5 MHz for a 10 Mbps PHY, and 25 MHz for a 100 Mbps PHY).
- 6) LCLK - Bus clock - Used by the BIU for synchronous accesses. Maximum frequency is 50 MHz for VL BUS mode, and 8.5 MHz for EISA slave DMA.

#### CSMA/CD Block

This is a 16-bit oriented block, with fully-independent Transmit and Receive logic. The data path in and out of the block consists of two 6-bit wide uni-directional FIFOs interfacing the

DMA block. The DMA port of the FIFO stores 32 bits exploiting the 32-bit data path into memory. The Control Path consists of a set of registers interfaced to the CPU via the BIU.

#### DMA Block

This block accesses packet memory on the CSMA/CD's behalf, fetching transmit data and storing received data. It interfaces the CSMA/CD Transmit and Receive FIFOs on one side, and the Arbiter block on the other. The data path is 32 bits wide.

The DMA machine is able to support full duplex operation. Independent receive and transmit counters are used. Transmit and receive cycles are alternated when simultaneous receive and transmit accesses are needed.

#### Arbiter Block

The Arbiter block sequences accesses to packet RAM requested by the BIU and by the DMA blocks. BIU requests represent pipelined CPU accesses to the Data Register, while DMA requests represent CSMA/CD data movement. The external memory devices used are 25ns 32kx8 SRAM. The cycle time for CPU consecutive accesses to the Data Path is 80ns/doubleword. This time includes arbitration and CSMA memory cycles.

The Arbiter is also responsible for controlling the nRWE0-nRWE3 lines as a function of the bytes being written. Read accesses are always 32 bits wide, and the Arbiter steers the appropriate byte(s) to the appropriate lanes as a function of the address.

The CPU Data Path consists of two uni-directional FIFOs mapped at the Data Register location. These FIFOs can be accessed in any

combination of bytes, word, or doublewords. The Arbiter will indicate 'Not Ready' whenever a cycle is initiated that cannot be satisfied by the present state of the FIFO.

The depth of the FIFOs will accommodate the worst case arbitration and byte access alignment pattern while still preserving the CPU cycle time when accessing the Data Register.

### **MMU Block**

The Hardware Memory Management Unit is similar to the LAN91C90's MMU. It does dynamic memory allocation and queuing of transmit and receive packets, and it determines the value of the transmit and receive interrupts as a function of the queues. The page size is still 2k, and with a maximum memory size of 128k the MMU uses 64x6 FIFOs. MIR and MCR values are interpreted in 512 byte units.

### **BIU Block**

The Bus Interface Unit can handle synchronous as well as asynchronous buses; different signals are used for each one. Transparent latches are added on the address path using rising nADS for latching.

When working with an asynchronous bus like ISA, the read and write operations are controlled by the edges of nRD and nWR. ARDY is used for notifying the system that it should extend the access cycle. The leading edge of ARDY is generated by the leading edge of nRD or nWR while the trailing edge of ARDY is controlled by the LAN91C100's internal clock and, therefore, is asynchronous to the bus.

In the synchronous VL Bus type mode, nCYCLE and LCLK are used for read and write operations. Completion of the cycle may be

determined by using nSRDY. nSRDY is controlled by LCLK and is synchronous to the bus.

Direct 32-bit access to the Data Path is supported by using the nDATACS input. By asserting nDATACS, external DMA-type of devices will bypass the BIU address decoders and can sequentially access memory with no CPU intervention. nDATACS accesses can be used in the DMA burst mode (nVLBUS=1) or in asynchronous cycles. These cycles MUST be 32-bit cycles. Please refer to the corresponding timing diagrams for details on these cycles.

### **MAC-PHY Interface**

Two separate interfaces are defined; one for the 10 Mbps bit rate interface and one for the MII 100 Mbps and 10 Mbps nibble rate interface. The 10 Mbps ENDEC interface comprises the signals used for interfacing Ethernet ENDECs. The 100 Mbps interface follows the MII draft standard for 100 Mbps 802.3 networks, and it is based on transferring nibbles between the MAC and the PHY.

For the MII interface, transmit data is clocked out using the TX25 clock input, while receive data is clocked in using RX25.

Switching between the ENDEC and MII interfaces is controlled by the MII Select bit in the Configuration Register. The MIISEL pin reflects the value of the bit and may be used to control external multiplexing logic.

### **MII Management Interface Block**

PHY management through the MII management interface is supported by the LAN91C100 by providing the means to drive a tri-statable data output, a clock, and reading an input. Timing and framing for each management command is generated by the CPU.



### Serial EEPROM Interface Block

This block is responsible for reading the serial EEPROM upon hardware reset (or equivalent command) and defining defaults for some key registers. A write operation is also implemented

by this block which, under CPU command, will program specific locations in the EEPROM. This block is an autonomous state machine, and it controls the LAN91C100's internal Data Bus during active operation.

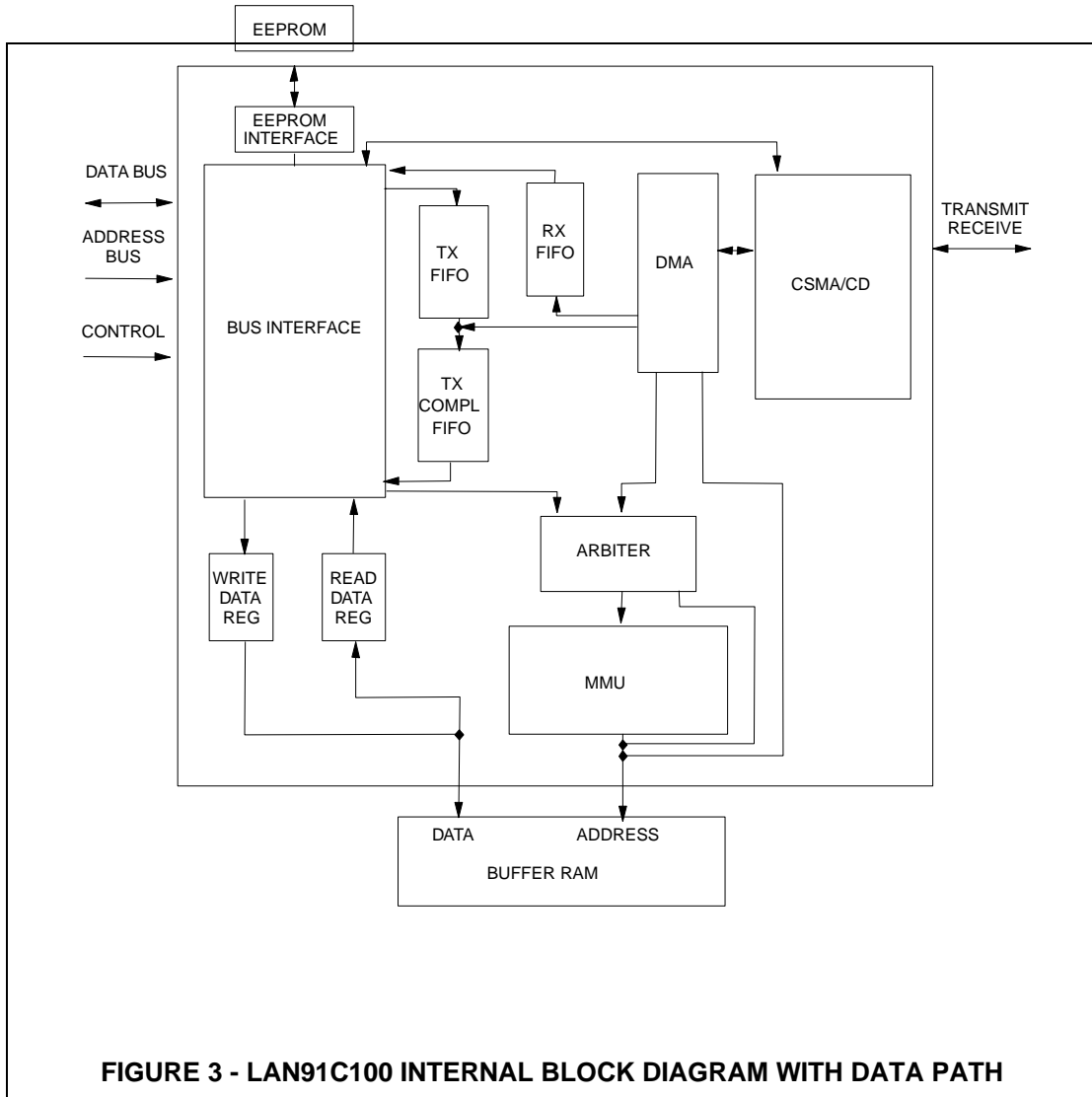


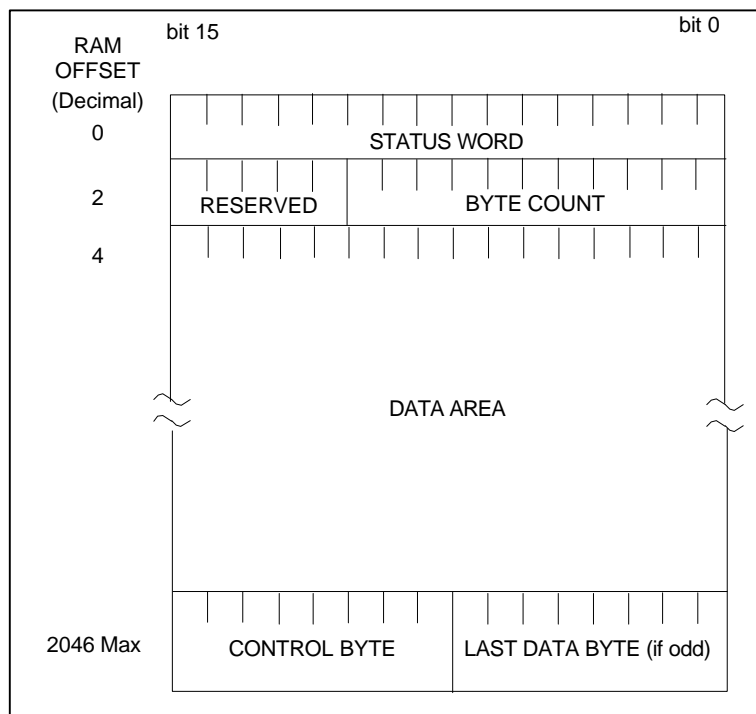
FIGURE 3 - LAN91C100 INTERNAL BLOCK DIAGRAM WITH DATA PATH

## DATA STRUCTURES AND REGISTERS

### PACKET FORMAT IN BUFFER MEMORY

The packet format in memory is similar for the Transmit and Receive areas. The first word is

reserved for the status word, the next word is used to specify the total number of bytes, and it is followed by the data area. The data area holds the packet itself.



**FIGURE 4 – DATA PACKET FORMAT**

	<b>TRANSMIT PACKET</b>	<b>RECEIVE PACKET</b>
STATUS WORD	Written by CSMA upon transmit completion (see Status Register)	Written by CSMA upon receive completion (see RX Frame Status Word)
BYTE COUNT	Written by CPU	Written by CSMA
DATA AREA	Written/modified by CPU	Written by CSMA
CONTROL BYTE	Written by CPU to control odd/even data bytes	Written by CSMA; also has odd/even bit

BYTE COUNT - Divided by two, it defines the total number of words, including the STATUS WORD, the BYTE COUNT WORD, the DATA AREA and the CONTROL BYTE.

The receive byte count always appears as even, the ODDFRM bit of the receive status word indicates if the low byte of the last word is relevant.

The transmit byte count least significant bit will be assumed 0 by the controller regardless of the value written in memory.

DATA AREA - The data area starts at offset 4 of the packet structure, and it can extend for up to 2043 bytes.

The data area contains six bytes of DESTINATION ADDRESS followed by six bytes of SOURCE ADDRESS, followed by a variable length number of bytes. On transmit, all bytes are provided by the CPU, including the source address. The LAN91C100 does not insert its own source address. On receive, all bytes are provided by the CSMA side.

The 802.3 Frame Length word (Frame Type in Ethernet) is not interpreted by the LAN91C100. It is treated transparently as data both for transmit and receive operations.

CONTROL BYTE - For transmit packets the CONTROL BYTE is written by the CPU as:

X	X	ODD	CRC	0	0	0	0
---	---	-----	-----	---	---	---	---

ODD If set, indicates an odd number of bytes, with the last byte being right before the CONTROL BYTE. If clear, the number of data bytes is even and the byte before the CONTROL BYTE is not transmitted.

CRC When set, CRC will be appended to the frame. This bit has only meaning if the NOCRC bit in the TCR is set.

For receive packets the CONTROL BYTE is written by the controller as:

0	1	ODD	0	0	0	0	0
---	---	-----	---	---	---	---	---

ODD If set, indicates an odd number of bytes, with the last byte being right before the CONTROL BYTE. If clear, the number of data

bytes is even and the byte before the CONTROL BYTE should be ignored.

## RECEIVE FRAME STATUS WORD

This word is written at the beginning of each receive frame in memory. It is not available as a register.

HIGH BYTE	ALGN ERR	BROD CAST	BAD CRC	ODD FRM	TOOLNG	TOO SHORT		
LOW BYTE	HASH VALUE						MULT CAST	
	5	4	3	2	1	0		

**ALGNERR** Frame had alignment error. When MII SEL=1 alignment error is set when BADCRC=1 and an odd number of nibbles were received between SFD and RX\_DV going inactive. When MII SEL=0 alignment error is set when BADCRC=1 and the number of bits received between SFD and the CRS going inactive is not an octet multiple.

**BROADCAST** Receive frame was broadcast.

**BADCRC** Frame had CRC error, or RX\_ER was asserted during reception.

**ODDFRM** This bit, when set, indicates that the received frame had an odd number of bytes.

**TOOLNG** Frame length was longer than 802.3 maximum size (1518 bytes on the cable).

**TOOSHORT** Frame length was shorter than 802.3 minimum size (64 bytes on the cable).

**HASH VALUE** Provides the hash value used to index the Multicast Registers. Can be used by receive routines to speed up the group address search. The hash value consists of the six most significant bits of the CRC calculated on the Destination Address, and maps into the 64 bit multicast table. Bits 5,4,3 of the hash value select a byte of the multicast table, while bits 2,1,0 determine the bit within the byte selected. Examples of the address mapping:

ADDRESS	HASH VALUE 5-0	MULTICAST TABLE BIT
ED 00 00 00 00 00	000 000	MT-0 bit 0
0D 00 00 00 00 00	010 000	MT-2 bit 0
01 00 00 00 00 00	100 111	MT-4 bit 7
2f 00 00 00 00 00	111 111	MT-7 bit 7

**MULTICAST** Receive frame was multicast. If hash value corresponds to a multicast table bit that is set, and the address was a multicast,

the packet will pass address filtering regardless of other filtering criteria.

## I/O SPACE

The base I/O space is determined by the IOS0-2 inputs and the EEPROM contents. To limit the I/O space requirements to 16 locations, the registers are assigned to different banks. The

last word of the I/O area is shared by all banks and can be used to change the bank in use. Registers are described using the following convention:

	OFFSET		NAME		TYPE		SYMBOL	
HIGH BYTE	bit 15	bit14	bit 13	bit 12	bit 11	bit 10	bit9	bit8
	X	X	X	X	X	X	X	X
LOW BYTE	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
	X	X	X	X	X	X	X	X

**OFFSET** Defines the address offset within the IOBASE where the register can be accessed at, provided the bank select has the appropriate value.

Some registers (like the Interrupt Ack., or like Interrupt Mask) are functionally described as two eight bit registers, in that case the offset of each one is independently specified.

The offset specifies the address of the even byte (bits 0-7) or the address of the complete word.

Regardless of the functional description, all registers can be accessed as doublewords, words or bytes.

The odd byte can be accessed using address (offset + 1).

The default bit values upon hard reset are **highlighted** below each register.

**Table 2 - Internal I/O Space Mapping**

	BANK0	BANK1	BANK2	BANK3
0	TCR	CONFIG	MMU COMMAND	MT0-1
2	EPH STATUS	BASE	PNR/ARR	MT2-3
4	RCR	IA0-1	FIFO PORTS	MT4-5
6	COUNTER	IA2-3	POINTER	MT6-7
8	MIR	IA4-5	DATA	MGMT
A	MCR	GENERAL PURPOSE	DATA	REVISION
C	RESERVED (0)	CONTROL	INTERRUPT	ERCV
E	BANK SELECT	BANK SELECT	BANK SELECT	BANK SELECT

A special BANK (BANK7) exists to support the addition of external registers.

## BANK SELECT REGISTER

OFFSET	NAME				TYPE		SYMBOL	
E	BANK SELECT REGISTER				READ/WRITE		BSR	
HIGH BYTE	0	0	1	1	0	0	1	1
	0	0	1	1	0	0	1	1
LOW BYTE						BS2	BS1	BS0
	X	X	X	X	X	0	0	0

BS2, BS1, BS0 Determine the bank presently in use. This register is always accessible and is used to select the register bank in use.

The upper byte always reads as 33h and can be used to help determine the I/O location of FEAST.

The BANK SELECT REGISTER is always accessible regardless of the value of BS0-2.

Note that the bank select register can be accessed as a doubleword at offset Ch, as a word at offset Eh, or as at offset Fh, however

a doubleword write to offset Ch will write the BANK SELECT REGISTER but will not write the registers Ch and Dh.

BANK 7 has no internal registers other than the BANK SELECT REGISTER itself. On valid cycles where BANK7 is selected (BS0=BS1=BS2=1), and A3=0, nCSOUT is activated to facilitate implementation of external registers.

Note: BANK7 does not exist in LAN91C9x devices. For backward S/W compatibility BANK7 accesses should be done if the Revision Control register indicates the device is LAN91C100.

**I/O SPACE - BANK0**

OFFSET	NAME	TYPE	SYMBOL
0	TRANSMIT CONTROL REGISTER	READ/WRITE	TCR

This register holds bits programmed by the CPU to control some of the protocol transmit options.

HIGH BYTE			EPH LOOP	STP SQET	FDUPLX	MON_ CSN		NOCRC
	X	X	0	0	0	0	X	0
LOW BYTE	PAD_EN					FORCOL	LOOP	TXENA
	0	X	X	X	X	0	0	0

**EPH\_LOOP** Internal loopback at the EPH block. Serial data is looped back when set. Defaults low. When EPH\_LOOP is high, the following transmit outputs are forced inactive: TXD0-3=0h, TXEN100=TXEN=0, TXD=1. The following external inputs are blocked: CRS=CRS100=0, COL=COL100=0, RX\_DV=RX\_ER=0.

**STP\_SQET** Stop transmission on SQET error. If set, stops and disables transmitter on SQE test error. Does not stop on SQET error and transmits next frame if clear. Defaults low.

**FDUPLX** When set it enables full duplex operation. This will cause frames to be received if they pass the address filter regardless of the source for the frame. When clear the node will not receive a frame sourced by itself.

**MON\_CSN** When set, the LAN91C100 monitors carrier while transmitting. It must see its own carrier by the end of the preamble. If it is not seen, or if carrier is lost during transmission, the transmitter aborts the frame

without CRC and turns itself off. When this bit is clear the transmitter ignores its own carrier. Defaults low.

**NOCRC** Does not append CRC to transmitted frames when set; allows software to insert the desired CRC. Defaults to 0 (CRC inserted).

**PAD\_EN** When set, the LAN91C100 will pad transmit frames shorter than 64 bytes with 00. Does not pad frames when reset.

**FORCOL** When set, the transmitter will force a collision by not deferring deliberately. After the collision this bit is reset automatically. This bit defaults low to normal operation.

**LOOP** Loopback. General purpose output port used to control the LBK pin. Typically used to put the PHY chip in loopback mode.

**TXENA** Transmit enabled when set. Transmit is disabled if clear. When the bit is cleared, the LAN91C100 will complete the current transmission before stopping. When stopping due to an error, this bit is automatically cleared.

## I/O SPACE - BANK 0

OFFSET	NAME	TYPE	SYMBOL
--------	------	------	--------

2	EPH STATUS REGISTER	READ ONLY	EPHSR
---	---------------------	-----------	-------

This register stores the status of the last frame transmitted. This register value, upon individual transmit packet completion, is stored as the first word in the memory area allocated to the packet. Packet interrupt processing should use the copy in memory as the register itself will be updated by subsequent packet transmissions. The register can be used for real time values (like TXENA and LINK OK). If TXENA is cleared the register holds the last packet completion status.

HIGH BYTE	TX_UNRN	LINK_OK	RX_OVRN	CTR_ROL	EXC_DEF	LOST_CARR	LATCOL	
	0	-nLNK Pin	0	0	0	0	0	X
LOW BYTE	TX_DEFR	LTX_BRD	SQET	16COL	LTX_MULT	MUL_COL	SNGL_COL	TX_SUC
	0	0	0	0	0	0	0	0

**TXUNRN** Transmit Under Run. Set if under run occurs, it also clears TXENA bit in TCR. Cleared by setting TXENA high. This bit should never be set under normal operation.

**LINK\_OK** General purpose input port driven by nLNK pin inverted. Typically used for LINK Test. A transition on the value of this bit generates an interrupt.

**RX\_OVRN** Upon FIFO overrun, the receiver asserts this bit and clears the FIFO. The receiver stays enabled. After a valid preamble has been detected on a subsequent frame, RX\_OVRN is de-asserted. The RX\_OVRN INT bit in the Interrupt Status Register will also be set and stay set until cleared by the CPU. Note that receive overruns could occur only if receive memory allocations fail.

**CTR\_ROL** Counter Roll Over. When set, one or more 4-bit counters have reached maximum count (15). Cleared by reading the ECR register.

**EXC\_DEF** Excessive Deferral. When set last/current transmit was deferred for more than 1518 \* 2 byte times. Cleared at the end of every packet sent.

**LOST\_CARR** Lost Carrier Sense. When set, indicates that Carrier Sense was not present at end of preamble. Valid only if MON\_CSN is enabled. This condition causes TXENA bit in TCR to be reset. Cleared by setting TXENA bit in TCR.

**LATCOL** Late collision detected on last transmit frame. If set, a late collision was detected (later than 64 byte times into the frame). When detected, the transmitter JAMs and turns itself off, clearing the TXENA bit in TCR. Cleared by setting TXENA in TCR.

**TX\_DEFR** Transmit Deferred. When set, carrier was detected during the first 6.4 μsec of the inter frame gap. Cleared at the end of every packet sent.

**LTX\_BRD** Last transmit frame was a broadcast.



Set if frame was broadcast. Cleared at the start of every transmit frame.

**SQET** Signal Quality Error Test. For 10 Mbps systems, the transmitter opens a 1.6  $\mu$ s window 0.8  $\mu$ s after transmission is completed and the receiver returns inactive. During this window, the transmitter expects to see the SQET signal from the transceiver. The absence of this signal is a 'Signal Quality Error' and is reported in this status bit. Transmission stops and EPH INT is set if STP\_SQET in the TCR is also set when SQET is set. This bit is cleared by setting TXENA high. The behavior of this bit for 100 Mbps is presently undefined.

**16COL** 16 collisions reached. Set when 16 collisions are detected for a transmit frame. TXENA bit in TCR is reset. Cleared when TXENA is set high.

**LTX\_MULT** Last transmit frame was a multicast. Set if frame was a multicast. Cleared at the start of every transmit frame.

**MULCOL** Multiple collision detected for the last transmit frame. Set when more than one collision was experienced. Cleared when TX\_SUC is high at the end of the packet being sent.

**SNGLCOL** Single collision detected for the last transmit frame. Set when a collision is detected. Cleared when TX\_SUC is high at the end of the packet being sent.

**TX\_SUC** Last transmit was successful. Set if transmit completes without a fatal error. This bit is cleared by the start of a new frame transmission or when TXENA is set high. Fatal errors are:

- 16 collisions
- SQET fail and STP\_SQET = 1
- FIFO Underrun
- Carrier lost and MON\_CSN = 1
- Late collision

**I/O SPACE - BANK 0**

OFFSET	NAME	TYPE	SYMBOL
4	RECEIVE CONTROL REGISTER	READ/WRITE	RCR

HIGH BYTE	SOFT RST	FILT CAR	0	0	0	0	STRIP CRC	RXEN
	0	0	0	0	0	0	0	0
LOW BYTE						ALMUL	PRMS	RX_ ABORT
	0	0	0	0	0	0	0	0

**SOFT\_RST** Software-activated Reset. Active high. Initiated by writing this bit high and terminated by writing the bit low. The LAN91C100's configuration is not preserved except for Configuration, Base, and IA0-5 Registers. EEPROM is not reloaded after software reset.

**FILT\_CAR** Filter Carrier. When set, filters leading edge of carrier sense for 12 bit times (3 nibble times). Otherwise recognizes a receive frame as soon as carrier sense is active. **(Does NOT filter RX\_DV on MII!)**

**STRIP\_CRC** When set, it strips the CRC on received frames. When clear, the CRC is stored in memory following the packet. Defaults low.

**RXEN** Enables the receiver when set. If cleared, completes receiving current frame and then goes idle. Defaults low on reset.

**ALMUL** When set, accepts all multicast frames (frames in which the first bit of DA is '1'). When clear accepts only the multicast frames that match the multicast table setting. Defaults low.

**PRMS** Promiscuous Mode. When set, receives all frames. **Does not receive its own transmission unless it is in full duplex!.**

**RX\_ABORT** This bit is set if a receive frame was aborted due to length longer than 2044 bytes. The frame will not be received. The bit is cleared by RESET or by the CPU writing it low.

**I/O SPACE - BANK 0**  
**OFFSET**

**NAME**

**TYPE**

**SYMBOL**

**6**

**COUNTER REGISTER**

**READ ONLY**

**ECR**

Counts four parameters for MAC statistics. When any counter reaches 15 an interrupt is issued. All counters are cleared when reading the register, and do not wrap around beyond 15.

HIGH BYTE	NUMBER OF EXC. DEFERRED TX				NUMBER OF DEFERRED TX			
	0	0	0	0	0	0	0	0
LOW BYTE	MULTIPLE COLLISION COUNT				SINGLE COLLISION COUNT			
	0	0	0	0	0	0	0	0

Each 4-bit counter is incremented every time the corresponding event, as defined in the EPH STATUS REGISTER bit description, occurs. Note that the counters can only increment once per enqueued transmit packet, never faster; limiting the rate of interrupts that can be generated by the counters. For example, if a packet is successfully transmitted after one collision, the SINGLE COLLISION COUNT field is incremented by one. If a packet experiences between two to 16 collisions, the MULTIPLE COLLISION COUNT field is incremented by one.

If a packet experiences deferral, the NUMBER OF DEFERRED TX field is incremented by one, even if the packet experienced multiple deferrals during its collision retries.

The COUNTER REGISTER facilitates maintaining statistics in the AUTO RELEASE mode where no transmit interrupts are generated on successful transmissions.

Reading the register in the transmit service routine will be enough to maintain statistics.

**I/O SPACE - BANK 0**  
**OFFSET**

**NAME**

**TYPE**

**SYMBOL**

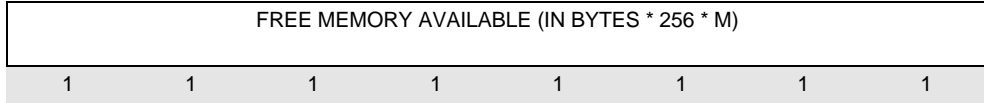
**8**

**MEMORY INFORMATION REGISTER**

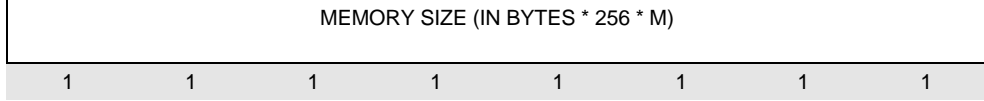
**READ ONLY**

**MIR**

HIGH  
 BYTE



LOW  
 BYTE



**FREE MEMORY AVAILABLE** This register can be read at any time to determine the amount of free memory. The register defaults to the **MEMORY SIZE** upon reset or upon the **RESET MMU** command.

All memory-related information is represented in  $256 * M$  byte units, where the multiplier  $M$  is determined by the **MCR** upper byte.

These registers default to **FFh**, which should be interpreted as **256**.

**MEMORY SIZE** - This register can be read to determine the total memory size.

**I/O SPACE - BANK 0**

OFFSET	NAME	TYPE	SYMBOL																
A	MEMORY CONFIGURATION REGISTER	Lower Byte - READ/WRITE Upper Byte - READ ONLY	MCR																
HIGH BYTE	<table border="1" style="width:100%; text-align:center;"> <tr> <td style="width:12.5%;"></td> <td style="width:12.5%;"></td> <td style="width:12.5%;"></td> <td style="width:12.5%;"></td> <td style="width:12.5%;">MEMORY SIZE MULTIPLIER (M)</td> <td style="width:12.5%;"></td> <td style="width:12.5%;"></td> <td style="width:12.5%;"></td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>1</td> <td>0</td> <td>1</td> <td>0</td> <td>1</td> </tr> </table>							MEMORY SIZE MULTIPLIER (M)				0	0	1	1	0	1	0	1
				MEMORY SIZE MULTIPLIER (M)															
0	0	1	1	0	1	0	1												
LOW BYTE	<table border="1" style="width:100%; text-align:center;"> <tr> <td colspan="8">MEMORY RESERVED FOR TRANSMIT (IN BYTES * 256 * M)</td> </tr> <tr> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> </tr> </table>			MEMORY RESERVED FOR TRANSMIT (IN BYTES * 256 * M)								0	0	0	0	0	0	0	0
MEMORY RESERVED FOR TRANSMIT (IN BYTES * 256 * M)																			
0	0	0	0	0	0	0	0												

**MEMORY RESERVED FOR TRANSMIT**  
 Programming this value allows the host CPU to reserve memory to be used later for transmit, limiting the amount of memory that receive packets can use up. When programmed for zero, the memory allocation between transmit and receive is completely dynamic. When programmed for a non-zero value, the allocation is dynamic if the free memory exceeds the programmed value, while receive allocation requests are denied if the free memory is less or equal to the programmed value. This register defaults to zero upon reset. It is not affected by the RESET MMU command.

The value written to the MCR is a reserved memory space IN ADDITION TO ANY MEMORY CURRENTLY IN USE. If the memory allocated for transmit plus the reserved space for transmit is required to be constant (rather than grow with transmit allocations), the CPU should update the value of this register after allocating or releasing memory.

The contents of MIR as well as the low byte of MCR are specified in 256 \* M bytes. The multiplier M is determined by bits 11, 10, and 9 as follows (Bits 11, 10 and 9 are read only bits used by the software driver to transparently run on different controllers of the LAN9000 family):

DEVICE	bit 11	Bit 10	bit 9	M	MAX MEMORY SIZE
LAN91C100	0	1	0	2	256*256*2=128k
LAN91C90	0	0	1	1	256*256*1=64k
FUTURE	0	1	1	4	256k
FUTURE	1	0	0	8	512k
FUTURE	1	0	1	16	1M

**I/O SPACE - BANK1**  
**OFFSET**

**NAME**

**TYPE**

**SYMBOL**

**0**

**CONFIGURATION REGISTER**

**READ/WRITE**

**CR**

The Configuration Register holds bits that define the adapter configuration and are not expected to change during run-time. This register is part of the EEPROM-saved setup.

HIGH BYTE	MII SELECT			NO WAIT		FULL STEP		AUI SELECT
	1	X	X	0	X	0	0	0
LOW BYTE	1	0	RESERVED			INT SEL1	INT SEL0	
	1	0	1	1	0	0	0	X

**MII SELECT** Used to select the network interface port. When set, the LAN91C100 will use its MII port and interface a PHY device at the nibble rate. When clear, the LAN91C100 will use its 10 Mbps ENDEC interface. This bit drives the MII SEL pin. Switching between ports should be done with transmitter and receiver disabled and no transmit/receive packets in progress.

**NO WAIT** When set, does not request additional wait states. An exception to this are accesses to the Data Register if not ready for a transfer. When clear, negates IOCHRDY for two to three clocks on any cycle to the LAN91C100.

**FULL STEP** This bit is a general purpose output port. Its inverse value drives pin nFSTEP and it is typically connected to SEL pin of the LAN83C694C. It can be used to select the signaling mode for the AUI, or as a general purpose non-volatile configuration pin. Defaults low.

**AUI SELECT** This bit is a general purpose output port. Its value drives pin AUISEL and it is typically connected to MODE1 pin of the LAN83C694C. It can be used to select AUI vs. 10BASE-T, or as a general purpose non-volatile configuration pin. Defaults low.

**INT SEL1-0** Used to select one out of four interrupt pins. The three unused interrupts are tristated.

INT SEL1	INT SEL0	PIN USED
0	0	INTR0
0	1	INTR1
1	0	INTR2
1	1	INTR3

**I/O SPACE - BANK1**  
**OFFSET**

**NAME**

**TYPE**

**SYMBOL**

**2**

**BASE ADDRESS REGISTER**

**READ/WRITE**

**BAR**

This register holds the I/O address decode option chosen for the LAN91C100. It is part of the EEPROM saved setup, and is not usually modified during run-time.

HIGH BYTE	A15	A14	A13	A9	A8	A7	A6	A5
	0	0	0	1	1	0	0	0
LOW BYTE	RESERVED							
	0	0	0	0	0	0	0	X

A15-A13 and A9-A5 These bits are compared against the I/O address on the bus to determine the IOBASE for the LAN91C100's registers. The 64k I/O space is fully decoded by the LAN91C100 down to a 16 location space,

therefore, the unspecified address lines A4, A10, A11 and A12 must be all zeros.

All bits in this register are loaded from the serial EEPROM. The I/O base decode defaults to 300h (namely, the high byte defaults to 18h).

**I/O SPACE - BANK1**

OFFSET	NAME	TYPE	SYMBOL
4 THROUGH 9	INDIVIDUAL ADDRESS REGISTERS	READ/WRITE	IAR

These registers are loaded starting at word location 20h of the EEPROM upon hardware reset or EEPROM reload. The registers can be modified by the software driver, but a STORE operation will not modify the EEPROM Individual Address contents. Bit 0 of Individual Address 0 register corresponds to the first bit of the address on the cable.

HIGH BYTE	ADDRESS 0
	0 0 0 0 0 0 0 0
LOW BYTE	ADDRESS 1
	0 0 0 0 0 0 0 0
HIGH BYTE	ADDRESS 2
	0 0 0 0 0 0 0 0
LOW BYTE	ADDRESS 3
	0 0 0 0 0 0 0 0
HIGH BYTE	ADDRESS 4
	0 0 0 0 0 0 0 0
LOW BYTE	ADDRESS 5
	0 0 0 0 0 0 0 0



**I/O SPACE - BANK1**  
**OFFSET**

**NAME**

**TYPE**

**SYMBOL**

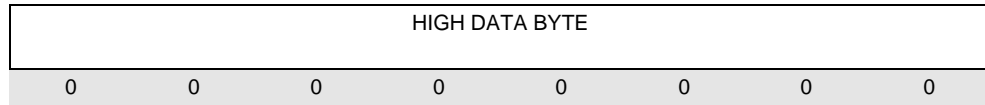
**A**

**GENERAL PURPOSE REGISTER**

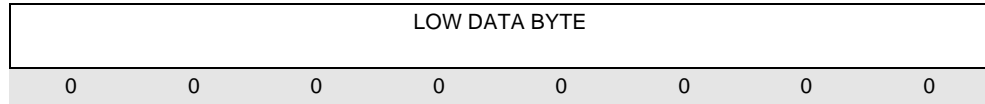
**READ/WRITE**

**GPR**

HIGH  
BYTE



LOW  
BYTE



This register can be used as a way of storing and retrieving non-volatile information in the EEPROM to be used by the software driver. The storage is word oriented, and the EEPROM word address to be read or written is specified using the six lowest bits of the Pointer Register.

This register can also be used to sequentially program the Individual Address area of the

EEPROM, that is normally protected from accidental Store operations.

This register will be used for EEPROM read and write only when the EEPROM SELECT bit in the Control Register is set. This allows generic EEPROM read and write routines that do not affect the basic setup of the LAN91C100.

**I/O SPACE - BANK1**

OFFSET	NAME				TYPE	SYMBOL		
C	CONTROL REGISTER				READ/WRITE	CTR		
HIGH BYTE	0	RCV_BAD	0		AUTO RELEASE			0
	0	0	0	X	0	X	X	0
LOW BYTE	LE ENABLE	CR ENABLE	TE ENABLE			EEPROM SELECT	RELOAD	STORE
	0	0	0	X	X	0	0	0

**RCV\_BAD** When set, bad CRC packets are received. When clear bad CRC packets do not generate interrupts and their memory is released.

**AUTO RELEASE** When set, transmit pages are released by transmit completion if the transmission was successful (when TX\_SUC is set). In that case there is no status word associated with its packet number, and successful packet numbers are not even written into the TX COMPLETION FIFO. A sequence of transmit packets will only generate an interrupt when the sequence is completely transmitted (TX EMPTY INT will be set), or when a packet in the sequence experiences a fatal error (TX INT will be set). Upon a fatal error TXENA is cleared and the transmission sequence stops. The packet number that failed is the present in the FIFO PORTS register, and its pages are not released, allowing the CPU to restart the sequence after corrective action is taken.

**LE ENABLE** Link Error Enable. When set it enables the LINK\_OK bit transition as one of the interrupts merged into the EPH INT bit. Defaults low (disabled). Writing this bit also serves as the acknowledge by clearing previous LINK interrupt conditions.

**CR ENABLE** Counter Roll over Enable. When set it enables the CTR\_ROL bit as one of the interrupts merged into the EPH INT bit. Defaults low (disabled).

**TE ENABLE** Transmit Error Enable. When set it enables Transmit Error as one of the interrupts merged into the EPH INT bit. Defaults low (disabled). Transmit Error is any condition that clears TXENA with TX\_SUC staying low as described in the EPHSR register.

**EEPROM SELECT** This bit allows the CPU to specify which registers the EEPROM RELOAD or STORE refers to. When high, the General Purpose Register is the only register read or written. When low, RELOAD reads Configuration, Base and Individual Address, and STORE writes the Configuration and Base registers.

**RELOAD** When set, it will read the EEPROM and update relevant registers with its contents. Clears upon completing the operation.

**STORE** When set, stores the contents of all relevant registers in the serial EEPROM. Clears upon completing the operation.

Note: When an EEPROM access is in progress the STORE and RELOAD bits will be read back as high. The remaining 14 bits of this register will be invalid. During this time attempted read/write operations, other than polling the

EEPROM status, will NOT have any effect on the internal registers. The CPU can resume accesses to the LAN91C100 after both bits are low. A worst case RELOAD operation initiated by RESET or by software takes less than 750 $\mu$ sec.

**I/O SPACE - BANK2**  
**OFFSET**

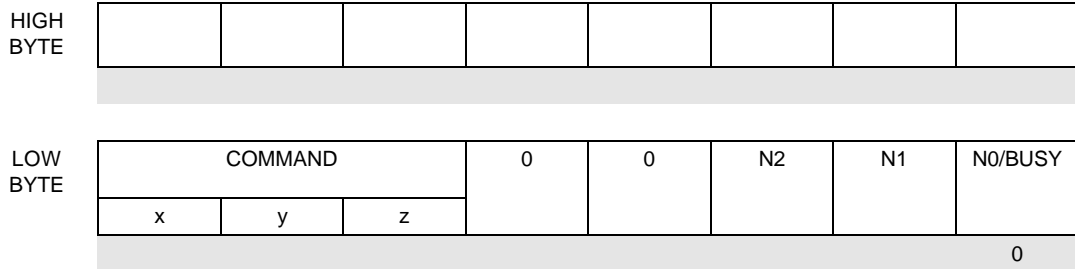
**NAME**

**TYPE**

**SYMBOL**

**0**                      **MMU COMMAND REGISTER**                      **WRITE ONLY**  
**BUSY Bit**  
**Readable**                      **MMUCR**

This register is used by the CPU to control the memory allocation, de-allocation, TX FIFO and RX FIFO control. The three command bits determine the command issued as described below:



**COMMAND SET**

xyz

- 000 0) NOOP - NO OPERATION
- 001 1) ALLOCATE MEMORY FOR TX - N2,N1,N0 defines the amount of memory requested as (value + 1) \* 256 bytes. Namely N2,N1,N0 = 1 will request 2 \* 256 = 512 bytes. A shift-based divide by 256 of the packet length yields the appropriate value to be used as N2,N1,N0. Immediately generates a completion code at the ALLOCATION RESULT REGISTER. Can optionally generate an interrupt on successful completion. N2,N1,N0 are ignored by the LAN91C100 but should be implemented in the LAN91C100's software drivers for LAN9000 compatibility.
- 010 2) RESET MMU TO INITIAL STATE - Frees all memory allocations, clears relevant interrupts, resets packet FIFO pointers.
- 011 3) REMOVE FRAME FROM TOP OF RX FIFO - To be issued after CPU has completed processing of present receive frame. This command removes the receive packet number from the RX FIFO and brings the next receive frame (if any) to the RX area (output of RX FIFO).
- 100 4) REMOVE AND RELEASE TOP OF RX FIFO - Like 3) but also releases all memory used by the packet presently at the RX FIFO output.

- 101 5) RELEASE SPECIFIC PACKET - Frees all pages allocated to the packet specified in the PACKET NUMBER REGISTER. Should not be used for frames pending transmission. Typically used to remove transmitted frames, after reading their completion status. Can be used following 3) to release receive packet memory in a more flexible way than 4).
- 110 6) ENQUEUE PACKET NUMBER INTO TX FIFO - This is the normal method of transmitting a packet just loaded into RAM. The packet number to be enqueued is taken from the PACKET NUMBER REGISTER.
- 111 7) RESET TX FIFOs - This command will reset both TX FIFOs--theTX FIFO holding the packet numbers awaiting transmission and the TX Completion FIFO. This command provides a mechanism for canceling packet transmissions, and reordering or bypassing the transmit queue. The RESET TX FIFOs command should only be used when the transmitter is disabled. Unlike the RESET MMU command, the RESET TX FIFOs does not release any memory.

Note 1: Bits N2,N1,N0 bits are ignored by the LAN91C100 but should be used for Command 0) to preserve software compatibility with the LAN91C92 and future devices. They should be zero for all other commands.

Note 2: When using the RESET TX FIFOS command, the CPU is responsible for releasing the memory associated with outstanding packets, or re-enqueuing them. Packet numbers in the completion FIFO can be read via the FIFO ports register before issuing the command.

Note 3: MMU commands releasing memory (commands 4 and 5) should only be issued if the corresponding packet number has memory allocated to it.

**COMMAND SEQUENCING**

A second allocate command (command 1) should not be issued until the present one has completed. Completion is determined by reading the FAILED bit of the allocation result register or through the allocation interrupt.

A second release command (commands 4, 5) should not be issued if the previous one is still being processed. The BUSY bit indicates that a release command is in progress. After issuing

command 5, the contents of the PNR should not be changed until BUSY goes low. After issuing command 4, command 3 should not be issued until BUSY goes low.

**BUSY BIT** Readable at bit 0 of the MMU command register address. When set indicates that MMU is still processing a release command. When clear, MMU has already completed last release command. BUSY and FAILED bits are set upon the trailing edge of command.

**I/O SPACE - BANK2**

OFFSET	NAME	TYPE	SYMBOL
2	PACKET NUMBER REGISTER	READ/WRITE	PNR

		PACKET NUMBER AT TX AREA					
0	0	0	0	0	0	0	0

PACKET NUMBER AT TX AREA - The value written into this register determines which packet number is accessible through the TX area. Some MMU commands use the number

stored in this register as the packet number parameter. This register is cleared by a RESET or a RESET MMU Command.

OFFSET	NAME	TYPE	SYMBOL
3	ALLOCATION RESULT REGISTER	READ ONLY	ARR

This register is updated upon an ALLOCATE MEMORY MMU command.

FAILED	ALLOCATED PACKET NUMBER					
1	0	0	0	0	0	0

**FAILED** A zero indicates a successful allocation completion. If the allocation fails the bit is set and only cleared when the pending allocation is satisfied. Defaults high upon reset and reset MMU command. For polling purposes, the ALLOC\_INT in the Interrupt Status Register should be used because it is synchronized to the read operation. Sequence:

- 1) Allocate Command
- 2) Poll ALLOC\_INT bit until set
- 3) Read Allocation Result Register

**ALLOCATED PACKET NUMBER** Packet number associated with the last memory allocation request. The value is only valid if the FAILED bit is clear.

Note: For software compatibility with future versions, the value read from the ARR after an allocation request is intended to be written into the PNR as is, without masking higher bits (provided FAILED = 0).

**I/O SPACE - BANK2****OFFSET****NAME****TYPE****SYMBOL****4****FIFO PORTS REGISTER****READ ONLY****FIFO**

This register provides access to the read ports of the Receive FIFO and the Transmit completion FIFO. The packet numbers to be processed by the interrupt service routines are read from this register.

HIGH BYTE	EMPTY		RX FIFO PACKET NUMBER						
	1	0	0	0	0	0	0	0	0
LOW BYTE	EMPTY		TX DONE PACKET NUMBER						
	1	0	0	0	0	0	0	0	0

**EMPTY** No receive packets queued in the RX FIFO. For polling purposes, uses the RCV\_INT bit in the Interrupt Status Register.

**TOP OF RX FIFO PACKET NUMBER** Packet number presently at the output of the RX FIFO. Only valid if EMPTY is clear. The packet is removed from the RX FIFO using MMU Commands 3) or 4).

**EMPTY** No transmit packets in completion queue. For polling purposes, uses the TX\_INT bit in the Interrupt Status Register.

**TX DONE PACKET NUMBER** Packet number presently at the output of the TX Completion FIFO. Only valid if EMPTY is clear. The packet is removed when a TX INT acknowledge is issued.

Note: For software compatibility with future versions, the value read from each FIFO register is intended to be written into the PNR as is, without masking higher bits (provided EMPTY and EMPTY = 0 respectively).

**I/O SPACE - BANK2**  
**OFFSET**

**NAME**

**TYPE**

**SYMBOL**

**6**

**POINTER REGISTER**

**READ/WRITE  
 NOT EMPTY is  
 a read only bit**

**PTR**

HIGH BYTE	RCV	AUTO INCR.	READ	ETEN	NOT EMPTY	POINTER HIGH		
	0	0	0	0	0	0	0	0
LOW BYTE	POINTER LOW							
	0	0	0	0	0	0	0	0

**POINTER REGISTER** The value of this register determines the address to be accessed within the transmit or receive areas. It will auto-increment on accesses to the data register when AUTO INCR. is set. The increment is by one for every byte access, by two for every word access, and by four for every double word access. When RCV is set, the address refers to the receive area and uses the output of RX FIFO as the packet number, when RCV is clear the address refers to the transmit area and uses the packet number at the Packet Number Register.

**READ** Determines the type of access to follow. If the READ bit is high, the operation intended is a read. If the READ bit is low, the operation is a write. Loading a new pointer value, with the READ bit high, generates a pre-fetch into the Data Register for read purposes.

Readback of the pointer will indicate the value of the address last accessed by the CPU (rather than the last pre-fetched). This allows any interrupt routine that uses the pointer, to save it and restore it without affecting the process being interrupted.

The Pointer Register should not be loaded until the CPU has verified that the NOT EMPTY bit is clear to ensure that the Data Register FIFO is empty. On reads, if IOCHRDY is not connected to the host, the Data Register should not be read before 370ns after the pointer was loaded to allow the Data Register FIFO to fill.

If the pointer is loaded using 8 bit writes, the low byte should be loaded first and the high byte last.

**ETEN** When set, enables EARLY Transmit underrun detection. Normal operation when clear.

**NOT EMPTY** When set, indicates that the Write Data FIFO is not empty yet. The CPU can verify that the FIFO is empty before loading a new pointer value. This is a read only bit.

Note: If AUTO INCR. is not set, the pointer must be loaded with an even value.



**I/O SPACE - BANK2**  
**OFFSET**

**NAME**

**TYPE**

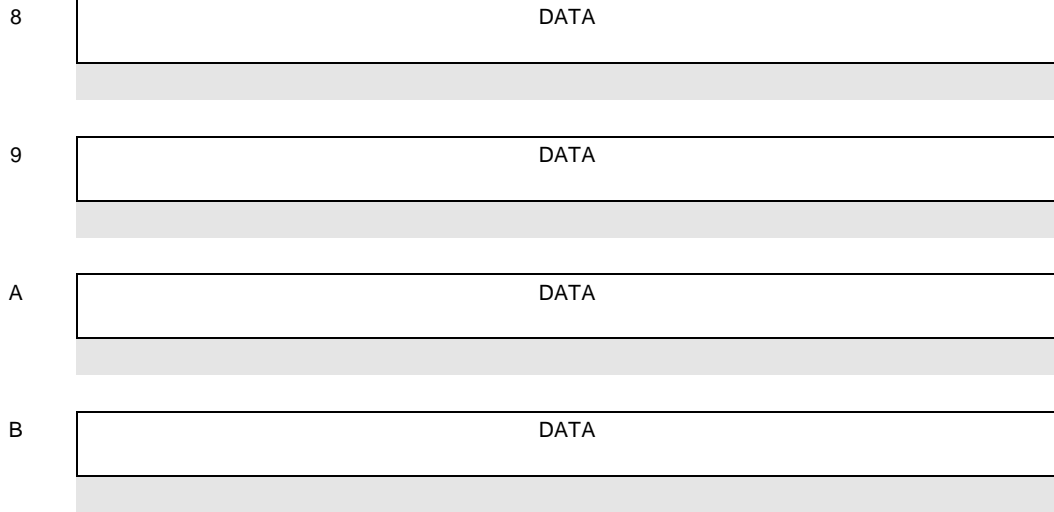
**SYMBOL**

**8 THROUGH Bh**

**DATA REGISTER**

**READ/WRITE**

**DATA**



**DATA REGISTER** Used to read or write the data buffer byte/word presently addressed by the pointer register.

Low or Data High registers. The order to and from the FIFO is preserved. Byte, word and dword accesses can be mixed on the fly in any order.

This register is mapped into two uni-directional FIFOs that allow moving words to and from the LAN91C100 regardless of whether the pointer address is even, odd or dword aligned. Data goes through the write FIFO into memory, and is pre-fetched from memory into the read FIFO. If byte accesses are used, the appropriate (next) byte can be accessed through the Data

This register is mapped into two consecutive word locations to facilitate double word move operations regardless of the actual bus width (16 or 32 bits). The DATA register is accessible at any address in the 8 through Ah range, while the number of bytes being transferred are determined by A1 and nBE0-nBE3. The FIFOs are 12 bytes each.

**I/O SPACE - BANK2**

OFFSET	NAME	TYPE	SYMBOL
<b>C</b>	<b>INTERRUPT STATUS REGISTER</b>	<b>READ ONLY</b>	<b>IST</b>

	ERCV INT	EPH INT	RX_OVRN INT	ALLOC INT	TX EMPTY INT	TX INT	RCV INT
X	0	0	0	0	1	0	0

OFFSET	NAME	TYPE	SYMBOL
<b>C</b>	<b>INTERRUPT ACKNOWLEDGE REGISTER</b>	<b>WRITE ONLY</b>	<b>ACK</b>

	ERCV INT		RX_OVRN INT		TX EMPTY INT	TX INT	

OFFSET	NAME	TYPE	SYMBOL
<b>D</b>	<b>INTERRUPT MASK REGISTER</b>	<b>READ/WRITE</b>	<b>MSK</b>

	ERCV INT	EPH INT	RX_OVRN INT	ALLOC INT	TX EMPTY INT	TX INT	RCV INT
X	0	0	0	0	0	0	0

This register can be read and written as a word or as two individual bytes.

The Interrupt Mask Register bits enable the appropriate bits when high and disable them when low. An enabled bit being set will cause a hardware interrupt.

**ERCV INT** Early receive interrupt. Set whenever a receive packet is being received, and the number of bytes received into memory

exceeds the value programmed as **ERCV THRESHOLD** (Bank 3, Offset Ch). **ERCV INT** stays set until acknowledged by writing the **INTERRUPT ACKNOWLEDGE REGISTER** with the **ERCV INT** bit set.

**EPH INT** Set when the Ethernet Protocol Handler section indicates one out of various possible special conditions. This bit merges exception type of interrupt sources, whose service time is not critical to the execution speed

of the low level drivers. The exact nature of the interrupt can be obtained from the EPH Status Register (EPHSR), and enabling of these sources can be done via the Control Register.

The possible sources are:

LINK - Link Test transition  
CTR\_ROL - Statistics counter roll over  
TXENA cleared - A fatal transmit error occurred forcing TXENA to be cleared.  
TX\_SUC will be low and the specific reason will be reflected by the bits:

TXUNRN - Transmit underrun  
SQET - SQE Error  
LOST CARR - Lost Carrier  
LATCOL - Late Collision  
16COL - 16 collisions

RX\_OVRN INT Set when the receiver overruns due to a failed memory allocation. The RX\_OVRN bit of the EPHSR will also be set, but if a new packet is received it will be cleared. The RX\_OVRN INT bit, however, latches the overrun condition for the purpose of being polled or generating an interrupt, and will only be cleared by writing the acknowledge register with the RX\_OVRN INT bit set.

ALLOC INT Set when an MMU request for TX pages allocation is completed. This bit is the complement of the FAILED bit in the ALLOCATION RESULT register. The ALLOC INT ENABLE bit should only be set following an allocation command, and cleared upon servicing the interrupt.

TX EMPTY INT Set if the TX FIFO goes empty, can be used to generate a single interrupt at the end of a sequence of packets enqueued for transmission. This bit latches the empty condition, and the bit will stay set until it is specifically cleared by writing the acknowledge register with the TX EMPTY INT bit set. If a

real time reading of the FIFO empty is desired, the bit should be first cleared and then read.

The TX EMPTY INT ENABLE should only be set after the following steps:

- a) a packet is enqueued for transmission
- b) the previous empty condition is cleared (acknowledged)

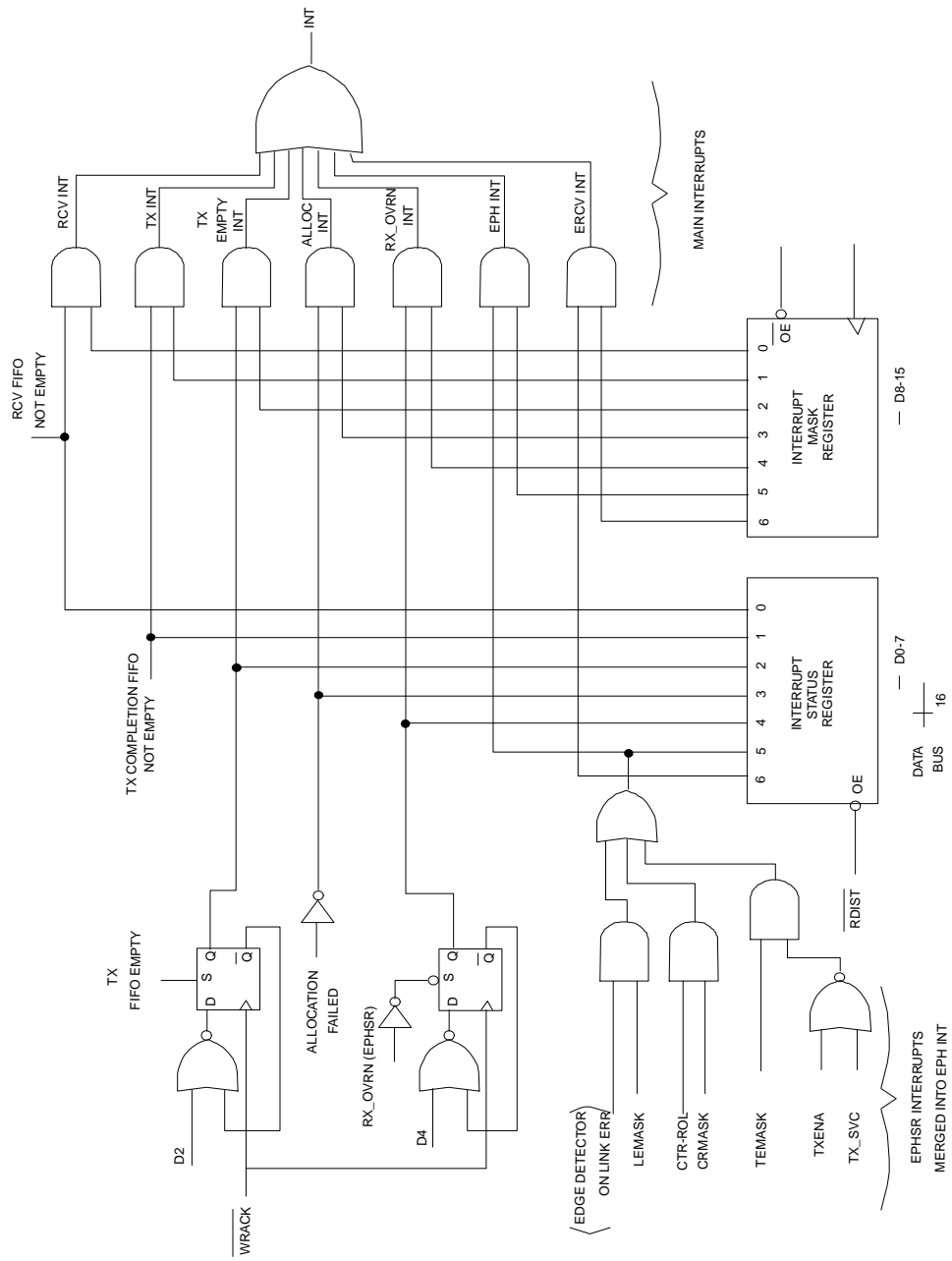
TX INT Set when at least one packet transmission was completed. The first packet number to be serviced can be read from the FIFO PORTS register. The TX INT bit is always the logic complement of the TEMPTY bit in the FIFO PORTS register. After servicing a packet number, its TX INT interrupt is removed by writing the Interrupt Acknowledge Register with the TX INT bit set.

RCV INT Set when a receive interrupt is generated. The first packet number to be serviced can be read from the FIFO PORTS register. The RCV INT bit is always the logic complement of the REMPTY bit in the FIFO PORTS register.

Note: If the driver uses AUTO RELEASE mode it should enable TX EMPTY INT as well as TX INT. TX EMPTY INT will be set when the complete sequence of packets is transmitted. TX INT will be set if the sequence stops due to a fatal error on any of the packets in the sequence.

Note: For edge triggered systems, the Interrupt Service Routine should clear the Interrupt Mask Register, and only enable the appropriate interrupts after the interrupt source is serviced (acknowledged).





**FIGURE 5 – INTERRUPT STRUCTURE**

I/O SPACE - BANK 3

OFFSET	NAME	TYPE	SYMBOL
0 THROUGH 7	MULTICAST TABLE	READ/WRITE	MT

LOW BYTE	MULTICAST TABLE 0		
	0 0 0 0 0 0 0 0		
HIGH BYTE	MULTICAST TABLE 1		
	0 0 0 0 0 0 0 0		
LOW BYTE	MULTICAST TABLE 2		
	0 0 0 0 0 0 0 0		
HIGH BYTE	MULTICAST TABLE 3		
	0 0 0 0 0 0 0 0		
LOW BYTE	MULTICAST TABLE 4		
	0 0 0 0 0 0 0 0		
HIGH BYTE	MULTICAST TABLE 5		
	0 0 0 0 0 0 0 0		
LOW BYTE	MULTICAST TABLE 6		
	0 0 0 0 0 0 0 0		
HIGH BYTE	MULTICAST TABLE 7		
	0 0 0 0 0 0 0 0		



The 64 bit multicast table is used for group address filtering. The hash value is defined as the six most significant bits of the CRC of the destination addresses. The three msb's determine the register to be used (MT0-7), while the other three determine the bit within the register.

If the appropriate bit in the table is set, the packet is received.

If the ALMUL bit in the RCR register is set, all multicast addresses are received regardless of the multicast table values.

Hashing is only a partial group addressing filtering scheme, but being the hash value available as part of the receive status word, the receive routine can reduce the search time significantly. With the proper memory structure, the search is limited to comparing only the multicast addresses that have the actual hash value in question.

**I/O SPACE - BANK3**  
**OFFSET**

**NAME**

**TYPE**

**SYMBOL**

**8**

**MANAGEMENT INTERFACE**

**READ/WRITE**

**MGMT**

HIGH  
 BYTE

0	0	1	1	0	0	1	1

LOW  
 BYTE

				MDOE	MCLK	MDI	MDO
0	0	1	1	0	0	MDI Pin	0

MDOE MII Management output enable. When high pin MDO is driven, when low pin MDO is tri-stated.

MCLK MII Management clock. The value of this bit drives the MDCLK pin.

MDI MII Management input. The value of the MDI pin is readable using this bit.

MDO MII Management output. The value of this bit drives the MDO pin.

The purpose of this interface, along with the corresponding pins, is to implement MII PHY management in software.

**I/O SPACE - BANK3**

OFFSET	NAME	TYPE	SYMBOL										
<b>A</b>	<b>REVISION REGISTER</b>	<b>READ ONLY</b>	<b>REV</b>										
HIGH BYTE	<table border="1" style="width:100%; height:20px;"> <tr> <td style="width:25%;">0</td> <td style="width:25%;">0</td> <td style="width:25%;">1</td> <td style="width:25%;">1</td> </tr> </table>			0	0	1	1						
0	0	1	1										
LOW BYTE	<table border="1" style="width:100%; height:20px;"> <tr> <td style="width:50%;">CHIP</td> <td style="width:50%;">REV</td> </tr> <tr> <td style="width:25%;">0</td> <td style="width:25%;">1</td> <td style="width:25%;">1</td> <td style="width:25%;">1</td> </tr> <tr> <td style="width:25%;">0</td> <td style="width:25%;">0</td> <td style="width:25%;">0</td> <td style="width:25%;">0</td> </tr> </table>			CHIP	REV	0	1	1	1	0	0	0	0
CHIP	REV												
0	1	1	1										
0	0	0	0										

CHIP Chip ID. Can be used by software drivers to identify the device used.

REV Revision ID. Incremented for each revision of a given device.

CHIP ID VALUE	DEVICE
3	LAN91C90/LAN91C92
7	LAN91C100

OFFSET	NAME	TYPE	SYMBOL												
<b>C</b>	<b>EARLY RCV REGISTER</b>	<b>READ/WRITE</b>	<b>ERCV</b>												
HIGH BYTE	<table border="1" style="width:100%; height:20px;"> <tr> <td style="width:25%;">0</td> <td style="width:25%;">0</td> <td style="width:25%;">1</td> <td style="width:25%;">1</td> </tr> </table>			0	0	1	1								
0	0	1	1												
LOW BYTE	<table border="1" style="width:100%; height:20px;"> <tr> <td style="width:25%;">RCV DISCRD</td> <td style="width:25%;"></td> <td style="width:25%;"></td> <td style="width:25%;">ERCV THRESHOLD</td> </tr> <tr> <td style="width:25%;">0</td> <td style="width:25%;">0</td> <td style="width:25%;">0</td> <td style="width:25%;">1</td> </tr> <tr> <td style="width:25%;">1</td> <td style="width:25%;">1</td> <td style="width:25%;">1</td> <td style="width:25%;">1</td> </tr> </table>			RCV DISCRD			ERCV THRESHOLD	0	0	0	1	1	1	1	1
RCV DISCRD			ERCV THRESHOLD												
0	0	0	1												
1	1	1	1												

RCV DISCRD Set to discard a packet being received.

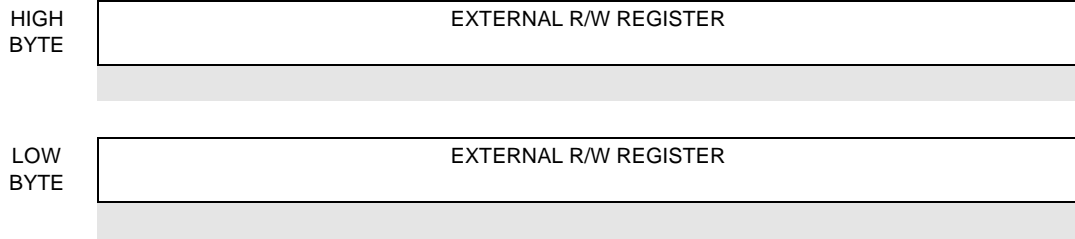
Whenever the number of bytes written in memory for the presently received packet exceeds the ERCV THRESHOLD, ERCV INT bit of the INTERRUPT STATUS REGISTER is set.

ERCV THRESHOLD Threshold for ERCV interrupt. Specified in 64 byte multiples.

**I/O SPACE - BANK 7**

OFFSET	NAME	TYPE	SYMBOL
<b>0 THROUGH 7</b>	<b>EXTERNAL REGISTERS</b>		

nCSOUT is driven low by the LAN91C100 when a valid access to the EXTERNAL REGISTER range occurs.



CYCLE	nCSOUT	LAN91C100 DATA BUS
AEN=0 A3=0 A4-15 matches I/O BASE BANK SELECT = 7	Driven low. Transparently latched on nADS rising edge.	Ignored on writes. Tri-stated on reads.
BANK SELECT = 4,5,6	High	Ignore cycle.
Otherwise	High	Normal LAN91C100 cycle.

## TYPICAL FLOW OF EVENTS FOR TRANSMIT

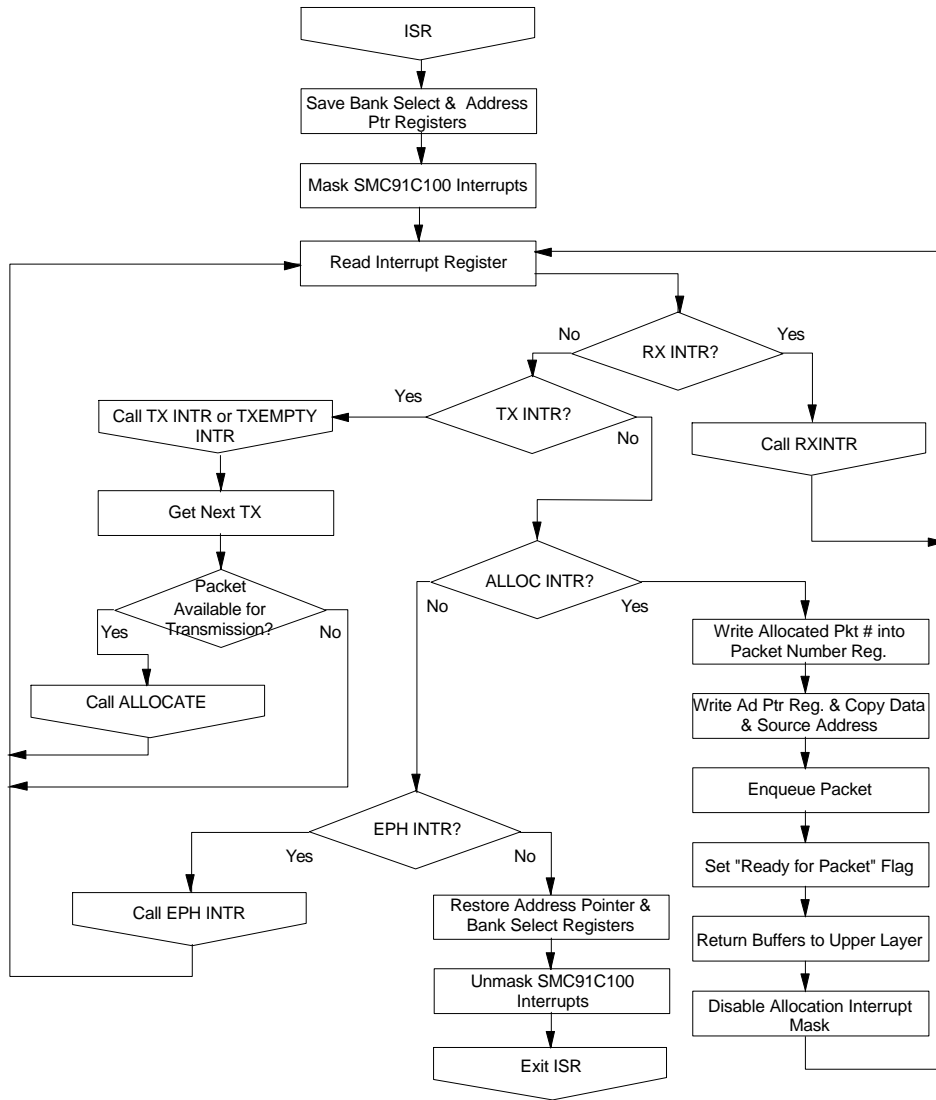
### S/W DRIVER

### MAC SIDE

- 1 ISSUE ALLOCATE MEMORY FOR TX - N BYTES - the MMU attempts to allocate N bytes of RAM.
  - 2 WAIT FOR SUCCESSFUL COMPLETION CODE - Poll until the ALLOC INT bit is set or enable its mask bit and wait for the interrupt. The TX packet number is now at the Allocation Result Register.
  - 3 LOAD TRANSMIT DATA - Copy the TX packet number into the Packet Number Register. Write the Pointer Register, then use a block move operation from the upper layer transmit queue into the Data Register.
  - 4 ISSUE "ENQUEUE PACKET NUMBER TO TX FIFO" - This command writes the number present in the Packet Number Register into the TX FIFO. The transmission is now enqueued. No further CPU intervention is needed until a transmit interrupt is generated.
  - 5
  - 6
  - 7 SERVICE INTERRUPT - Read Interrupt Status Register. If it is a transmit interrupt, read the TX Done Packet Number from the Fifo Ports Register. Write the packet number into the Packet Number Register. The corresponding status word is now readable from memory. If status word shows successful transmission, issue RELEASE packet number command to free up the memory used by this packet. Remove packet number from completion FIFO by writing TX INT Acknowledge Register.
- The enqueued packet will be transferred to the MAC block as a function of TXENA (n TCR) bit and of the deferral process state.
- Upon transmit completion the first word in memory is written with the status word. The packet number is moved from the TX FIFO into the TX completion FIFO. Interrupt is generated by the TX completion FIFO being not empty.

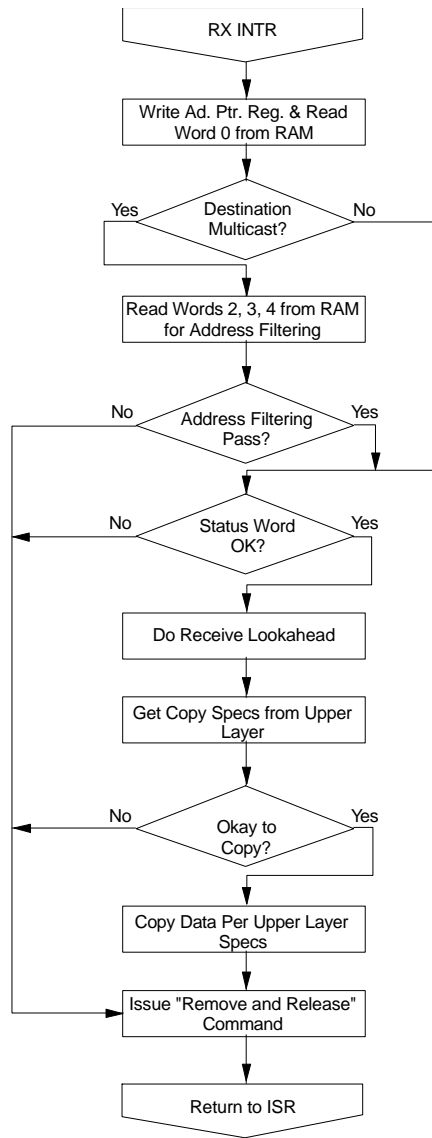
## TYPICAL FLOW OF EVENTS FOR RECEIVE

S/W DRIVER	MAC SIDE
1 ENABLE RECEPTION - By setting the RXEN bit.	
2	A packet is received with matching address. Memory is requested from MMU. A packet number is assigned to it. Additional memory is requested if more pages are needed.
3	The internal DMA logic generates sequential addresses and writes the receive words into memory. The MMU does the sequential to physical address translation. If overrun, packet is dropped and memory is released.
4	When the end of packet is detected, the status word is placed at the beginning of the receive packet in memory. Byte count is placed at the second word. If the CRC checks correctly the packet number is written into the RX FIFO. The RX FIFO being not empty causes RCV INT (interrupt) to be set. If CRC is incorrect the packet memory is released and no interrupt will occur.
5 SERVICE INTERRUPT - Read the Interrupt Status Register and determine if RCV INT is set. The next receive packet is at receive area. (Its packet number can be read from the FIFO Ports Register). The software driver can process the packet by accessing the RX area, and can move it out to system memory if desired. When processing is complete the CPU issues the REMOVE AND RELEASE FROM TOP OF RX command to have the MMU free up the used memory and packet number.	

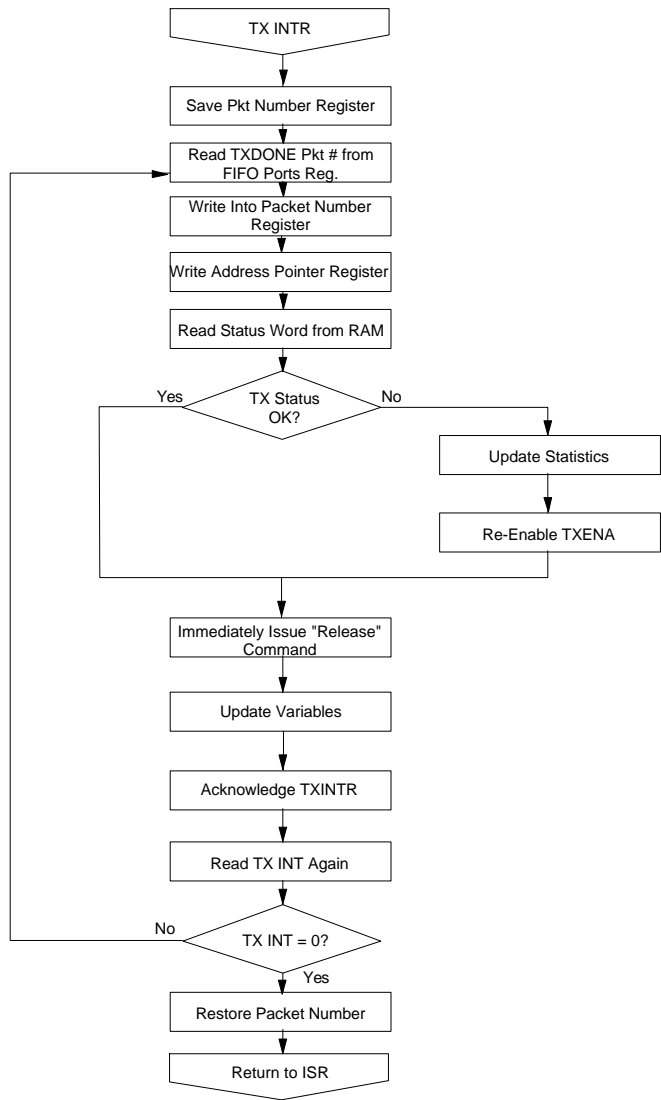


**FIGURE 6 – INTERRUPT SERVICE ROUTINE**

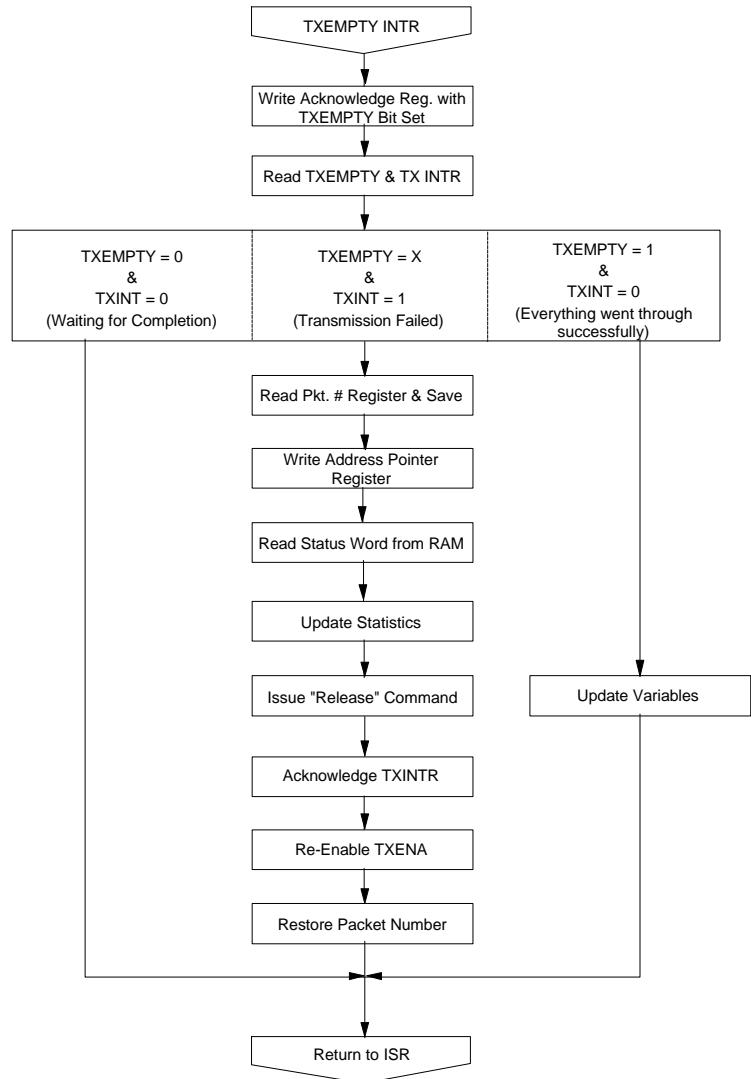




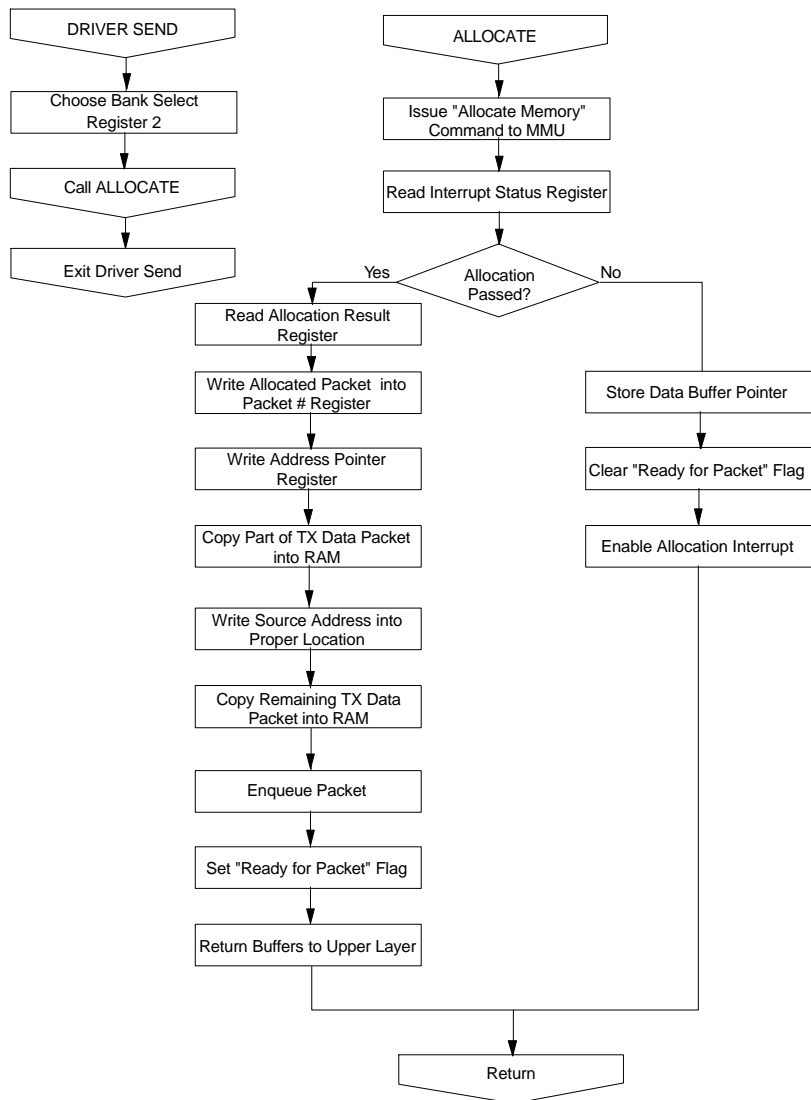
**FIGURE 7 – RX INTR**



**FIGURE 8 – TX INTR**



**FIGURE 9 – TXEMPTY INTR  
(Assumes Auto Release Selected)**



**FIGURE 10 – DRIVER SEND AND ALLOCATE ROUTINES**

## MEMORY PARTITIONING

Unlike other controllers, the LAN91C100 does not require a fixed memory partitioning between transmit and receive resources. The MMU allocates and de-allocates memory upon different events. An additional mechanism allows the CPU to prevent the receive process from starving the transmit memory allocation.

Memory is always requested by the side that needs to write into it, that is: the CPU for transmit or the MAC for receive. The CPU can control the number of bytes it requests for transmit but it cannot determine the number of bytes the receive process is going to demand. Furthermore, the receive process requests will be dependent on network traffic, in particular on the arrival of broadcast and multicast packets that might not be for the node, and that are not subject to upper layer software flow control.

In order to prevent unwanted traffic from using too much memory, the CPU can program a "memory reserved for transmit" parameter. If the free memory falls below the "memory reserved for transmit" value, MMU requests from the MAC block will fail and the packets will overrun and be ignored. Whenever enough memory is released, packets can be received again. If the reserved value is too large, the node might lose data which is an abnormal condition. If the value is kept at zero, memory allocation is handled on first-come first-served basis for the entire memory capacity.

Note that with the memory management built into the LAN91C100, the CPU can dynamically program this parameter. For instance, when the driver does not need to enqueue transmissions, it can allow more memory to be allocated for receive (by reducing the value of the reserved memory). Whenever the driver needs to burst transmissions it can reduce the receive memory allocation. The driver program the parameter as a function of the following variables:

- 1) Free memory (read only register)

- 2) Memory size (read only register)

The reserved memory value can be changed on the fly. If the MEMORY RESERVED FOR TX value is increased above the FREE MEMORY, receive packets in progress are still received, but no new packets are accepted until the FREE MEMORY increases above the MEMORY RESERVED value.

## INTERRUPT GENERATION

The interrupt strategy for the transmit and receive processes is such that it does not represent the bottleneck in the transmit and receive queue management between the software driver and the controller. For that purpose there is no register reading necessary before the next element in the queue (namely transmit or receive packet) can be handled by the controller. The transmit and receive results are placed in memory.

The receive interrupt will be generated when the receive queue (FIFO of packets) is not empty and receive interrupts are enabled. This allows the interrupt service routine to process many receive packets without exiting, or one at a time if the ISR just returns after processing and removing one.

There are two types of transmit interrupt strategies:

- 1) One interrupt per packet.
- 2) One interrupt per sequence of packets.

The strategy is determined by how the transmit interrupt bits and the AUTO RELEASE bit are used.

TX INT bit - Set whenever the TX completion FIFO is not empty.

TX EMPTY INT bit - Set whenever the TX FIFO is empty.



AUTO RELEASE - When set, successful transmit packets are not written into completion FIFO, and their memory is released automatically.

1) One interrupt per packet: enable TX INT, set AUTO RELEASE=0. The software driver can find the completion result in memory and process the interrupt one packet at a time. Depending on the completion code the driver will take different actions. Note that the transmit process is working in parallel and other transmissions might be taking place. The LAN91C100 is virtually queuing the packet numbers and their status words.

In this case, the transmit interrupt service routine can find the next packet number to be serviced by reading the TX DONE PACKET NUMBER at the FIFO PORTS register. This eliminates the need for the driver to keep a list of packet numbers being transmitted. The numbers are queued by the LAN91C100 and provided back to the CPU as their transmission completes.

2) One interrupt per sequence of packets: Enable TX EMPTY INT and TX INT, set AUTO RELEASE=1. TX EMPTY INT is generated only after transmitting the last packet in the FIFO.

TX INT will be set on a fatal transmit error allowing the CPU to know that the transmit process has stopped and therefore the FIFO will not be emptied.

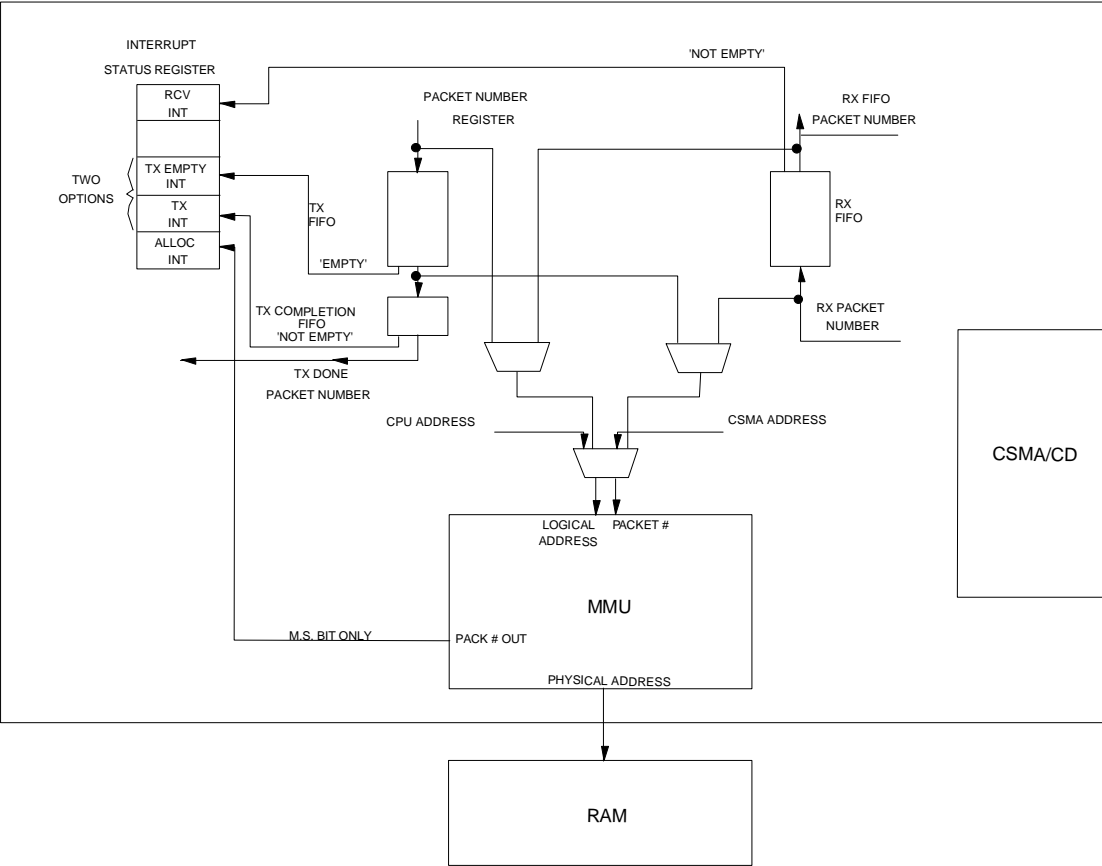
This mode has the advantage of a smaller CPU overhead, and faster memory de-allocation. Note that when AUTO RELEASE=1 the CPU is not provided with the packet numbers that completed successfully.

Note: The pointer register is shared by any process accessing the LAN91C100 memory. In order to allow processes to be interruptable, the interrupting process is responsible for reading the pointer value before modifying it, saving it, and restoring it before returning from the interrupt.

Typically there would be three processes using the pointer:

- 1) Transmit loading (sometimes interrupt driven)
- 2) Receive unloading (interrupt driven)
- 3) Transmit Status reading (interrupt driven).

1) and 3) also share the usage of the Packet Number Register. Therefore saving and restoring the PNR is also required from interrupt service routines.



**FIGURE 11 – INTERRUPT GENERATION FOR TRANSMIT, RECEIVE, MMU**

## BOARD SETUP INFORMATION

The following parameters are obtained from the EEPROM as board setup information:

ETHERNET INDIVIDUAL ADDRESS  
I/O BASE ADDRESS  
10BASE-T or AUI INTERFACE  
MII or ENDEC INTERFACE  
INTERRUPT LINE SELECTION

REGISTER	EEPROM WORD ADDRESS
Configuration Register	IOS Value * 4
Base Register	(IOS Value * 4) + 1

INDIVIDUAL ADDRESS                      20-22 hex

All the above mentioned values are read from the EEPROM upon hardware reset. Except for the INDIVIDUAL ADDRESS, the value of the IOS switches determines the offset within the EEPROM for these parameters, in such a way that many identical boards can be plugged into the same system by just changing the IOS jumpers.

In order to support a software utility based installation, even if the EEPROM was never programmed, the EEPROM can be written using the LAN91C100. One of the IOS combination is associated with a fixed default value for the key parameters (I/O BASE, INTERRUPT) that can always be used regardless of the EEPROM based value being programmed. This value will be used if all IOS pins are left open or pulled high.

The EEPROM is arranged as a 64 x 16 array. The specific target device is the 9346 1024-bit Serial EEPROM. All EEPROM accesses are done in words. All EEPROM addresses in the spec are specified as word addresses.

If IOS2-0 = 7, only the INDIVIDUAL ADDRESS is read from the EEPROM. Currently assigned values are assumed for the other registers. These values are default if the EEPROM read operation follows hardware reset.

The EEPROM SELECT bit is used to determine the type of EEPROM operation: a) normal or b) general purpose register.

a) NORMAL EEPROM OPERATION - EEPROM SELECT bit = 0

On EEPROM read operations (after reset or after setting RELOAD high) the CONFIGURATION REGISTER and BASE REGISTER are updated with the EEPROM values at locations defined by the IOS2-0 pins. The INDIVIDUAL ADDRESS registers are updated with the values stored in the INDIVIDUAL ADDRESS area of the EEPROM.

On EEPROM write operations (after setting the STORE bit) the values of the CONFIGURATION REGISTER and BASE REGISTER are written in the EEPROM locations defined by the IOS2-0 pins.

The three least significant bits of the CONTROL REGISTER (EEPROM SELECT, RELOAD and STORE) are used to control the EEPROM. Their values are not stored nor loaded from the EEPROM.

b) GENERAL PURPOSE REGISTER -  
EEPROM SELECT bit = 1

On EEPROM read operations (after setting RELOAD high) the EEPROM word address defined by the POINTER REGISTER 6 least significant bits is read into the GENERAL PURPOSE REGISTER.

On EEPROM write operations (after setting the STORE bit) the value of the GENERAL

PURPOSE REGISTER is written at the EEPROM word address defined by the POINTER REGISTER 6 least significant bits.

RELOAD and STORE are set by the user to initiate read and write operations respectively. Polling the value until read low is used to determine completion. When an EEPROM access is in progress the STORE and RELOAD bits of CTR will readback as both bits high. No other bits of FEAST can be read or written until the EEPROM operation completes and both bits are clear. This mechanism is also valid for reset initiated reloads.

Note: If no EEPROM is connected to the LAN91C900, for example for some embedded applications, the ENEEP pin should be grounded and no accesses to the EEPROM will be attempted. Configuration, Base, and Individual Address assume their default values upon hardware reset and the CPU is responsible for programming them for their final value.

IOS2-0	WORD ADDRESS	16 BITS	
		Configuration	Base
000	0h	CONFIGURATION REG.	
	1h		BASE REG.
001	4h	CONFIGURATION REG.	
	5h		BASE REG.
010	8h	CONFIGURATION REG.	
	9h		BASE REG.
011	Ch	CONFIGURATION REG.	
	Dh		BASE REG.
100	10h	CONFIGURATION REG.	
	11h		BASE REG.
101	14h	CONFIGURATION REG.	
	15h		BASE REG.
110	18h	CONFIGURATION REG.	
	19h		BASE REG.
XXX	20h		IA0-1
	21h		IA2-3
	22h		IA4-5

**FIGURE 12 – 64 X 16 SERIAL EEPROM MAP**

## APPLICATION CONSIDERATIONS

The LAN91C100 is envisioned to fit a few different bus types. This section describes the basic guidelines, system level implications and sample configurations for the most relevant bus types. All applications are based on buffered architectures with a private SRAM bus.

### FAST ETHERNET SLAVE ADAPTER

Slave non-intelligent board implementing 100 Mbps and 10 Mbps speeds.

Adapter requires:

- a) LAN91C100 Fast Ethernet Controller
- b) Four SRAMs (32k x 8 - 25ns)
- c) Serial EEPROM (93C46)
- d) 10 Mbps ENDEC and transceiver chip

- e) 100 Mbps MII compliant PHY
- f) Some bus specific glue logic

Target systems:

- a) VL Local Bus 32 bit systems) VL Local Bus 32 bit systems) VL Local Bus 32 bit systems) VL Local Bus 32 bit systems
- b) High-end ISA machines
- c) EISA 32 bit slave

### VL Local Bus 32 Bit Systems

On VL Local Bus and other 32 bit embedded systems, the LAN91C100 is accessed as a 32 bit peripheral in terms of the bus interface. All registers except the DATA REGISTER will be accessed using byte or word instructions. Accesses to the DATA REGISTER could use byte, word, or dword instructions.

**Table 3 - VL Local Bus Signal Connections**

VL BUS SIGNAL	LAN91C100 SIGNAL	NOTES
A2-A15	A2-A15	Address bus used for I/O space and register decoding, latched by nADS rising edge, and transparent on nADS low time
M/nIO	AEN	Qualifies valid I/O decoding - enabled access when low. This signal is latched by nADS rising edge and transparent on nADS low time
W/nR	W/nR	Direction of access. Sampled by the LAN91C100 on first rising clock that has nCYCLE active. High on writes, low on reads.
nRDYRTN	nRDYRTN	Ready return. Direct connection to VL bus.
nLRDY	nSRDY and some logic	nSRDY has the appropriate functionality and timing to create the VL nLRDY except that nLRDY behaves like an open drain output most of the time.



**Table 3 - VL Local Bus Signal Connections**

<b>VL BUS SIGNAL</b>	<b>LAN91C100 SIGNAL</b>	<b>NOTES</b>																																													
LCLK	LCLK	Local Bus Clock. Rising edges used for synchronous bus interface transactions.																																													
nRESET	RESET	Connected via inverter to the LAN91C100.																																													
nBE0 nBE1 nBE2 nBE3	nBE0 nBE1 nBE2 nBE3	Byte enables. Latched transparently by nADS rising edge.																																													
nADS	nADS, nCYCLE	Address Strobe is connected directly to the VL bus. nCYCLE is created typically by using nADS delayed by one LCLK.																																													
IRQn	INTR0-INTR3	Typically uses the interrupt lines on the ISA edge connector of VL bus.																																													
D0-D31	D0-D31	<p>32 bit data bus. The bus byte(s) used to access the device are a function of nBE0-nBE3:</p> <table border="1"> <thead> <tr> <th><b>BE0</b></th> <th><b>BE1</b></th> <th><b>nBE2</b></th> <th><b>BE3</b></th> <th></th> </tr> </thead> <tbody> <tr> <td></td> <td></td> <td><b>2</b></td> <td></td> <td></td> </tr> <tr> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>Double word access</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>1</td> <td>Low word access</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> <td>0</td> <td>High word access</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>1</td> <td>Byte 0 access</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>1</td> <td>Byte 1 access</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> <td>1</td> <td>Byte 2 access</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> <td>0</td> <td>Byte 3 access</td> </tr> </tbody> </table> <p>□ Not used = tri-state on reads, ignored on writes. Note that nBE2 and nBE3 override the value of A1, which is tied low in this application.</p>	<b>BE0</b>	<b>BE1</b>	<b>nBE2</b>	<b>BE3</b>				<b>2</b>			0	0	0	0	Double word access	0	0	1	1	Low word access	1	1	0	0	High word access	0	1	1	1	Byte 0 access	1	0	1	1	Byte 1 access	1	1	0	1	Byte 2 access	1	1	1	0	Byte 3 access
<b>BE0</b>	<b>BE1</b>	<b>nBE2</b>	<b>BE3</b>																																												
		<b>2</b>																																													
0	0	0	0	Double word access																																											
0	0	1	1	Low word access																																											
1	1	0	0	High word access																																											
0	1	1	1	Byte 0 access																																											
1	0	1	1	Byte 1 access																																											
1	1	0	1	Byte 2 access																																											
1	1	1	0	Byte 3 access																																											
nLDEV	nLDEV	nLDEV is a totem pole output. nLDEV is active on valid decodes of A15-A4 and AEN=0.																																													
<b>UNUSED PINS</b>																																															
VCC	nRD, nWR																																														
GND	A1, nVLBUS																																														
OPEN	nDATACS																																														

## HIGH-END ISA MACHINES

On ISA machines, the LAN91C100 is accessed as a 16 bit peripheral. No support for XT (8 bit peripheral) is provided. The signal connections are listed in the following table:

**Table 4 - High-End ISA Machines Signal Connections**

ISA BUS SIGNAL	LAN91C100 SIGNAL	NOTES																
A1-A15	A1-A15	Address bus used for I/O space and register decoding																
AEN	AEN	Qualifies valid I/O decoding - enabled access when low																
nIORD	nRD	I/O Read strobe - asynchronous read accesses. Address is valid before leading edge																
nIOWR	nWR	I/O Write strobe - asynchronous write access. Address is valid before leading edge. Data is latched on trailing edge																
IOCHRDY	ARDY	This signal is negated on leading nRD, nWR if necessary. It is then asserted on CLK rising edge after the access condition is satisfied.																
RESET	RESET																	
A0	nBE0																	
nSBHE	nBE1																	
IRQn	INTR0-INTR3																	
D0-D15	D0-D15	16 bit data bus. The bus byte(s) used to access the device are a function of nBE0 and nBE1: <table style="margin-left: auto; margin-right: auto; border-collapse: collapse;"> <thead> <tr> <th style="padding: 2px;">nBE0</th> <th style="padding: 2px;">nBE1</th> <th style="padding: 2px;">D0-D7</th> <th style="padding: 2px;">D8-D15</th> </tr> </thead> <tbody> <tr> <td style="padding: 2px;">0</td> <td style="padding: 2px;">0</td> <td style="padding: 2px;">Lower</td> <td style="padding: 2px;">Upper</td> </tr> <tr> <td style="padding: 2px;">0</td> <td style="padding: 2px;">1</td> <td style="padding: 2px;">Lower</td> <td style="padding: 2px;">Not Used</td> </tr> <tr> <td style="padding: 2px;">1</td> <td style="padding: 2px;">0</td> <td style="padding: 2px;">Not Used</td> <td style="padding: 2px;">Upper</td> </tr> </tbody> </table> <p>Not used = tri-state on reads, ignored on writes.</p>	nBE0	nBE1	D0-D7	D8-D15	0	0	Lower	Upper	0	1	Lower	Not Used	1	0	Not Used	Upper
nBE0	nBE1	D0-D7	D8-D15															
0	0	Lower	Upper															
0	1	Lower	Not Used															
1	0	Not Used	Upper															
nIOCS16	nLDEV buffered	nLDEV is a totem pole output. Must be buffered using an open collector driver. nLDEV is active on valid decodes of A15-A4 and AEN=0.																

**Table 4 - High-End ISA Machines Signal Connections**

<b>ISA BUS SIGNAL</b>	<b>LAN91C100 SIGNAL</b>	<b>NOTES</b>
<b>UNUSED PINS</b>		
VCC	nBE2, nBE3, nCYCLE, W/nR nRDYRTN	No upper word access.
GND	LCLK, nADS	
OPEN	D16-D31, nDATACS, nVLBUS	

**EISA 32 BIT SLAVE EISA 32 BIT SLAVE EISA 32 BIT SLAVE EISA 32 BIT SLAVE EISA 32 BIT SLAVE EISA 32 BIT SLAVE EISA 32 BIT SLAVE**

On EISA, the LAN91C100 is accessed as a 32 bit I/O slave, along with a Slave DMA type "C" data path option. As an I/O slave, the LAN91C100 uses asynchronous accesses. In creating nRD and nWR inputs, the timing information is externally derived from nCMD edges. Given that the access will be at least 1.5 to 2 clocks (more than 180ns at least) there is no need to negate EXRDY, simplifying the EISA interface implementation. As a DMA

Slave, the LAN91C100 accepts burst transfers, and is able to sustain the peak rate of one doubleword every BCLK. Doubleword alignment is assumed for DMA transfers. Up to 3 extra bytes in the beginning and at the end of the transfer should be moved by the CPU using I/O accesses to the Data Register. The LAN91C100 will sample EXRDY and postpone DMA cycles if the memory cycle solicits wait states.

**Table 5 - EISA 32 Bit Slave Signal Connections**

<b>EISA BUS SIGNAL</b>	<b>LAN91C100 SIGNAL</b>	<b>NOTES</b>
LA2-15	A2-A15	Address bus used for I/O space and register decoding, latched by nADS (nSTART) trailing edge.
M/nIO AEN	AEN	Qualifies valid I/O decoding - enabled access when low. These signals are externally ORed. Internally the AEN pin is latched by nADS rising edge and transparent while nADS is low.
Latched W-R combined with nCMD	nRD	I/O Read strobe - Asynchronous read accesses. Address is valid before its leading edge. Must not be active during DMA bursts if DMA is supported.
Latched W-R combined with nCMD	nWR	I/O Write strobe - Asynchronous write access. Address is valid before leading edge. Data latched on trailing edge. Must not be active during DMA bursts if DMA is supported.
nSTART	nADS	Address strobe is connected to EISA nSTART.
RESDRV	RESET	
nBE0, nBE1, nBE2, nBE3	nBE0, nBE1, nBE2, nBE3	Byte enables. Latched on nADS rising edge.
IRQn	INTR0-INTR3	Interrupts used as active high edge triggered.

**Table 5 - EISA 32 Bit Slave Signal Connections**

<b>EISA BUS SIGNAL</b>	<b>LAN91C100 SIGNAL</b>	<b>NOTES</b>																																													
D0-D31	D0-D31	<p>32 bit data bus. The bus byte(s) used to access the device are a function of nBE0-nBE3:</p> <table border="0"> <thead> <tr> <th>nBE</th> <th>nBE</th> <th>nBE</th> <th>nBE</th> <th></th> </tr> <tr> <th>0</th> <th>1</th> <th>2</th> <th>3</th> <th></th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>Double word access</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>1</td> <td>Low word access</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> <td>0</td> <td>High word access</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>1</td> <td>Byte 0 access</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>1</td> <td>Byte 1 access</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> <td>1</td> <td>Byte 2 access</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> <td>0</td> <td>Byte 3 access</td> </tr> </tbody> </table> <p>Not used = tri-state on reads, ignored on writes. Note that nBE2 and nBE3 override the value of A1, which is tied low in this application. Other combinations of nBE are not supported by the LAN91C100. S/W drivers are not anticipated to generate them.</p>	nBE	nBE	nBE	nBE		0	1	2	3		0	0	0	0	Double word access	0	0	1	1	Low word access	1	1	0	0	High word access	0	1	1	1	Byte 0 access	1	0	1	1	Byte 1 access	1	1	0	1	Byte 2 access	1	1	1	0	Byte 3 access
nBE	nBE	nBE	nBE																																												
0	1	2	3																																												
0	0	0	0	Double word access																																											
0	0	1	1	Low word access																																											
1	1	0	0	High word access																																											
0	1	1	1	Byte 0 access																																											
1	0	1	1	Byte 1 access																																											
1	1	0	1	Byte 2 access																																											
1	1	1	0	Byte 3 access																																											
nEX32 nNOWS (optional additional logic)	nLDEV	<p>nLDEV is a totem pole output. nLDEV is active on valid decodes of the LAN91C100's pins A15-A4 and AEN=0. nNOWS is similar to nLDEV except that it should go inactive on nSTART rising. nNOWS can be used to request compressed cycles (1.5 BCLK long, nRD/nWR will be 1/2 BCLK wide).</p>																																													
<b>THE FOLLOWING SIGNALS SUPPORT SLAVE DMA TYPE "C" BURST CYCLES</b>																																															
BCLK	LCLK	EISA Bus Clock. Data transfer clock for DMA bursts.																																													
nDAK<n>	nDATACS	DMA Acknowledge. Active during Slave DMA cycles. Used by the LAN91C100 as nDATACS direct access to data path.																																													
nIORC	W/nR	Indicates the direction and timing of the DMA cycles. High during LAN91C100 writes; low during LAN91C100 reads.																																													
nIOWC	nCYCLE	Indicates slave DMA writes.																																													
nEXRDY	nRDYRTN	EISA bus signal indicating whether a slave DMA cycle will take place on the next BCLK rising edge, or should be postponed. nRDYRTN is used as an input in the slave																																													

		DMA mode to bring in EXRDY.
--	--	-----------------------------

**Table 5 - EISA 32 Bit Slave Signal Connections**

<b>EISA BUS SIGNAL</b>	<b>LAN91C100 SIGNAL</b>	<b>NOTES</b>
<b>UNUSED PINS</b>		
VCC	nVLBUS	
GND	A1	
OPEN		

## OPERATIONAL DESCRIPTION

### MAXIMUM GUARANTEED RATINGS\*

Operating Temperature Range .....	0°C to +70°C
Storage Temperature Range .....	-55°C to +150°C
Lead Temperature Range (soldering, 10 seconds) .....	+325°C
Positive Voltage on any pin, with respect to Ground .....	$V_{CC} + 0.3V$
Negative Voltage on any pin, with respect to Ground .....	-0.3V
Maximum $V_{CC}$ .....	+7V

\*Stresses above those listed above could cause permanent damage to the device. This is a stress rating only and functional operation of the device at any other condition above those indicated in the operation sections of this specification is not implied.

Note: When powering this device from laboratory or system power supplies, it is important that the Absolute Maximum Ratings not be exceeded or device failure can result. Some power supplies exhibit voltage spikes on their outputs when the AC power is switched on or off. In addition, voltage transients on the AC power line may appear on the DC output. If this possibility exists, it is suggested that a clamp circuit be used.

### DC ELECTRICAL CHARACTERISTICS ( $T_A = 0^\circ\text{C} - 70^\circ\text{C}$ , $V_{CC} = +5.0\text{ V} \pm 10\%$ )

PARAMETER	SYMBOL	MIN	TYP	MAX	UNITS	COMMENTS
<b>I Type Input Buffer</b>						
Low Input Level	$V_{ILI}$			0.8	V	TTL Levels
High Input Level	$V_{IHI}$	2.0			V	
<b>IS Type Input Buffer</b>						
Low Input Level	$V_{ILIS}$			0.8	V	Schmitt Trigger
High Input Level	$V_{IHIS}$	2.2			V	Schmitt Trigger
Schmitt Trigger Hysteresis	$V_{HYS}$		250		mV	
<b>I<sub>CLK</sub> Input Buffer</b>						
Low Input Level	$V_{ILCK}$			0.4	V	
High Input Level	$V_{IHCK}$	3.0			V	



PARAMETER	SYMBOL	MIN	TYP	MAX	UNITS	COMMENTS
<b>Input Leakage</b> (All I and IS buffers except pins with pullups/pulldowns)						
Low Input Leakage	I <sub>IL</sub>	-10		+10	μA	V <sub>IN</sub> = 0
High Input Leakage	I <sub>IH</sub>	-10		+10	A	V <sub>IN</sub> = V <sub>CC</sub>
<b>IP Type Buffers</b>						
Input Current	I <sub>IL</sub>	-150	-75		μA	V <sub>IN</sub> = 0
<b>ID Type Buffers</b>						
Input Current	I <sub>IH</sub>		+75	+150	μA	V <sub>IN</sub> = V <sub>CC</sub>
<b>O4 Type Buffer</b>						
Low Output Level	V <sub>OL</sub>			0.4	V	I <sub>OL</sub> = 4 mA
High Output Level	V <sub>OH</sub>	2.4			V	I <sub>OH</sub> = -2 mA
Output Leakage	I <sub>OL</sub>	-10		+10	μA	V <sub>IN</sub> = 0 to V <sub>CC</sub>
<b>I/O4 Type Buffer</b>						
Low Output Level	V <sub>OL</sub>			0.4	V	I <sub>OL</sub> = 4 mA
High Output Level	V <sub>OH</sub>	2.4			V	I <sub>OH</sub> = -2 mA
Output Leakage	I <sub>OL</sub>	-10		+10	μA	V <sub>IN</sub> = 0 to V <sub>CC</sub>
<b>O12 Type Buffer</b>						
Low Output Level	V <sub>OL</sub>			0.5	V	I <sub>OL</sub> = 12 mA
High Output Level	V <sub>OH</sub>	2.4			V	I <sub>OH</sub> = -6 mA
Output Leakage	I <sub>OL</sub>	-10		+10	μA	V <sub>IN</sub> = 0 to V <sub>CC</sub>
<b>O16 Type Buffer</b>						
Low Output Level	V <sub>OL</sub>			0.5	V	I <sub>OL</sub> = 16 mA
High Output Level	V <sub>OH</sub>	2.4			V	I <sub>OH</sub> = -8 mA
Output Leakage	I <sub>OL</sub>	-10		+10	μA	V <sub>IN</sub> = 0 to V <sub>CC</sub>

PARAMETER	SYMBOL	MIN	TYP	MAX	UNITS	COMMENTS
<b>OD16 Type Buffer</b>						
Low Output Level	$V_{OL}$			0.5	V	$I_{OL} = 16 \text{ mA}$
Output Leakage	$I_{OL}$	-10		+10	$\mu\text{A}$	$V_{IN} = 0 \text{ to } V_{CC}$
<b>O24 Type Buffer</b>						
Low Output Level	$V_{OL}$			0.5	V	$I_{OL} = 24 \text{ mA}$
High Output Level	$V_{OH}$	2.4			V	$I_{OH} = -12 \text{ mA}$
Output Leakage	$I_{OL}$	-10		+10	$\mu\text{A}$	$V_{IN} = 0 \text{ to } V_{CC}$
<b>I/O24 Type Buffer</b>						
Low Output Level	$V_{OL}$			0.5	V	$I_{OL} = 24 \text{ mA}$
High Output Level	$V_{OH}$	2.4			V	$I_{OH} = -12 \text{ mA}$
Output Leakage	$I_{OL}$	-10		+10	$\mu\text{A}$	$V_{IN} = 0 \text{ to } V_{CC}$
Supply Current Active	$I_{CC}$		60	95	mA	All outputs open.
Supply Current Standby	$I_{CSBY}$		8		mA	

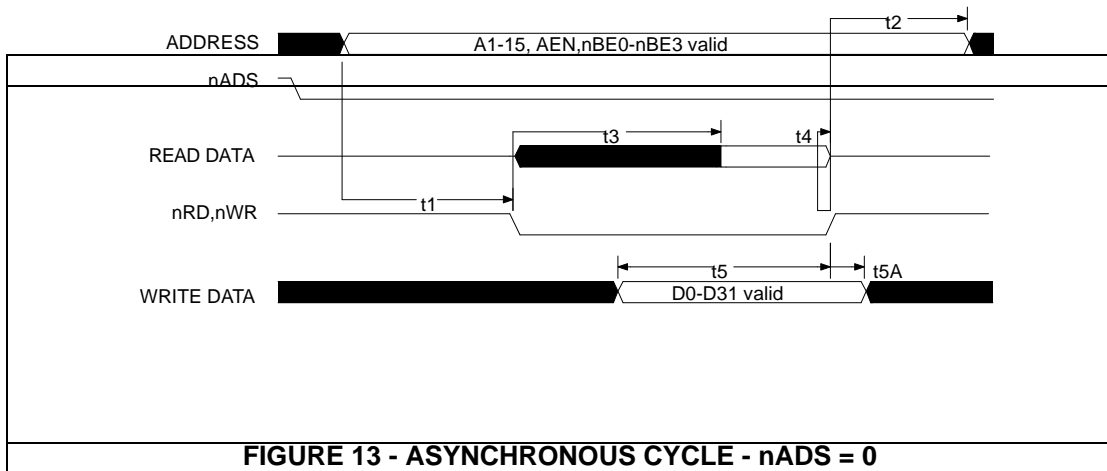
CAPACITANCE  $T_A = 25^\circ\text{C}$ ;  $f_c = 1\text{MHz}$ ;  $V_{CC} = 5\text{V}$

PARAMETER	SYMBOL	LIMITS			UNIT	TEST CONDITION
		MIN	TYP	MAX		
Clock Input Capacitance	$C_{IN}$			20	pF	All pins except pin under test tied to AC ground
Input Capacitance	$C_{IN}$			10	pF	
Output Capacitance	$C_{OUT}$			20	pF	

#### CAPACITIVE LOAD ON OUTPUTS

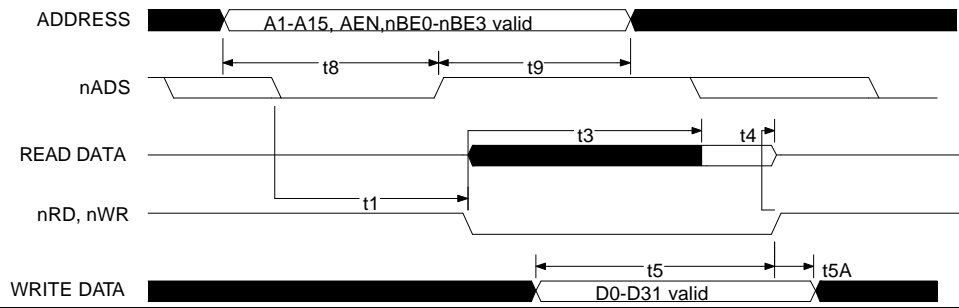
nARDY, D0-D31 (non VLBUS)	240 pF
D0-D31 in VLBUS	45 pF
All other outputs	45 pF

## TIMING DIAGRAMS



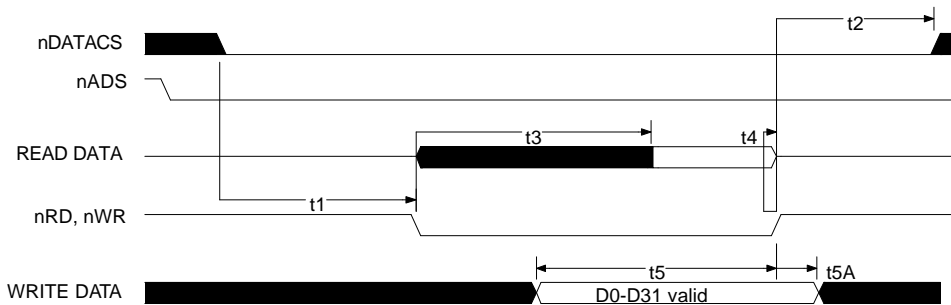
	PARAMETER	MIN	TYP	MAX	UNITS
t1	A1-A15, AEN, nBE0-nBE3 Valid and nADS Low Setup to nRD, nWR Active	25			ns
t2	A1-A15, AEN, nBE0-nBE3 Hold After nRD, nWR Inactive (Assuming nADS Tied Low)	20			ns
t3	nRD Low to Valid Data			40	ns
t4	nRD High to Data Floating			30	ns
t5	Data Setup to nWR Inactive	30			ns
t5A	Data Hold After nWR Inactive	5			ns

**FIGURE 14 - ASYNCHRONOUS CYCLE - USING nADS**



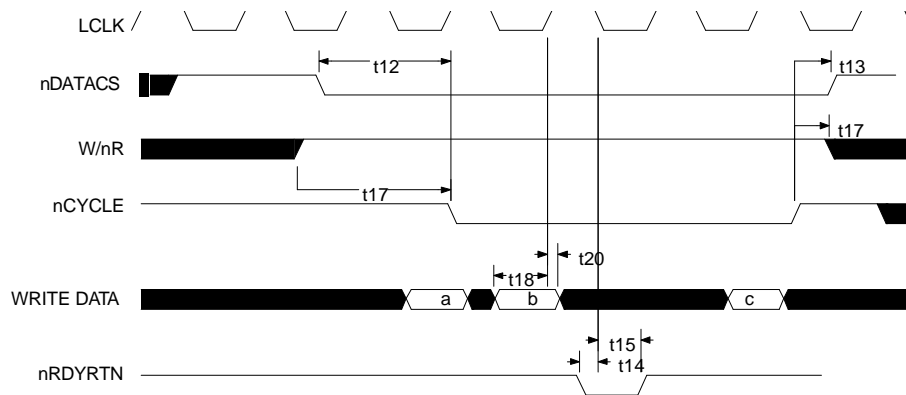
	PARAMETER	MIN	TYP	MAX	UNITS
t1	A1-A15, AEN, nBE0-nBE3 Valid and nADS Low Setup to nRD, nWR Active	25			ns
t3	nRD Low to Valid Data			40	ns
t4	nRD High to Data Floating			30	ns
t5	Data Setup to nWR Inactive	30			ns
t5A	Data Hold After nWR Inactive	5			ns
t8	A1-A15, AEN, nBE0-nBE3 Setup to nADS Rising	10			ns
t9	A1-A15, AEN, nBE0-nBE3 Hold after nADS Rising	15			ns

**FIGURE 15 - ASYNCHRONOUS CYCLE - nADS = 0  
(nDATACS Used to Select Data Register; Must Be 32 Bit Access)**



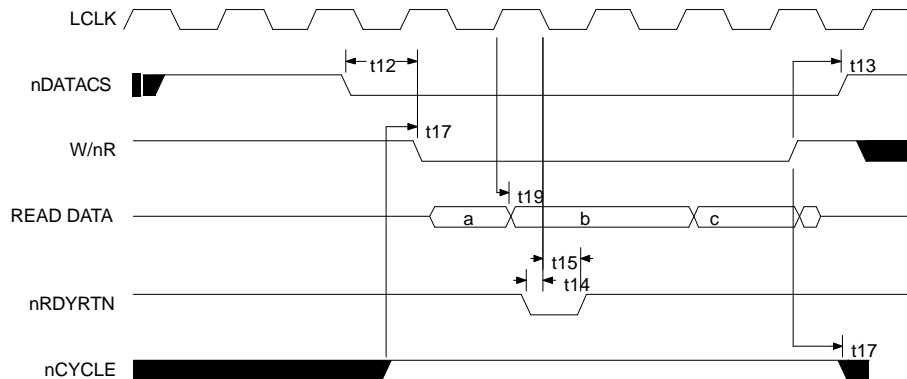
	PARAMETER	MIN	TYP	MAX	UNITS
t1	A1-A15, AEN, nBE0-nBE3 Valid and nADS Low Setup to nRD, nWR Active	25			ns
t2	A1-A15, AEN, nBE0-nBE3 Hold After nRD, nWR Inactive (Assuming nADS Tied Low)	20			ns
t3	nRD Low to Valid Data			40	ns
t4	nRD High to Data Floating			30	ns
t5	Data Setup to nWR Inactive	30			ns
t5A	Data Hold After nWR Inactive	5			ns

**FIGURE 16 - BURST WRITE CYCLES - nVLBUS = 1**



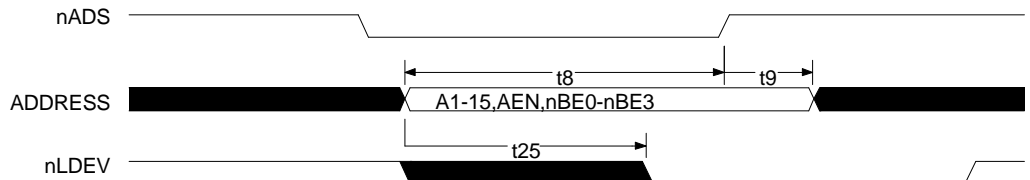
	PARAMETER	MIN	TYP	MAX	UNITS
t12	nDATACS Setup to Either nCYCLE or W/nR Falling	60			ns
t13	nDATACS Hold after Either nCYCLE or W/nR Rising	30			ns
t14	nRDYRTN Setup to LCLK Falling	15			ns
t15	nRDYRTN Hold after LCLK Falling	2			ns
t17	nCYCLE High and W/nR High Overlap	50			ns
t18	Data Setup to LCLK Rising (Write)	13			ns
t20	Data Hold from LCLK Rising (Write)	5			ns

**FIGURE 17 - BURST READ CYCLES - nVLBUS = 1**



	PARAMETER	MIN	TYP	MAX	UNITS
t12	nDATACS Setup to Either nCYCLE or W/nR Falling	60			ns
t13	nDATACS Hold after Either nCYCLE or W/nR Rising	30			ns
t14	nRDYRTN Setup to LCLK Falling	15			ns
t15	nRDYRTN Hold after LCLK Falling	2			ns
t17	nCYCLE High and W/nR High Overlap	50			ns
t19	Data Delay from LCLK Rising (Read)	5		38	ns

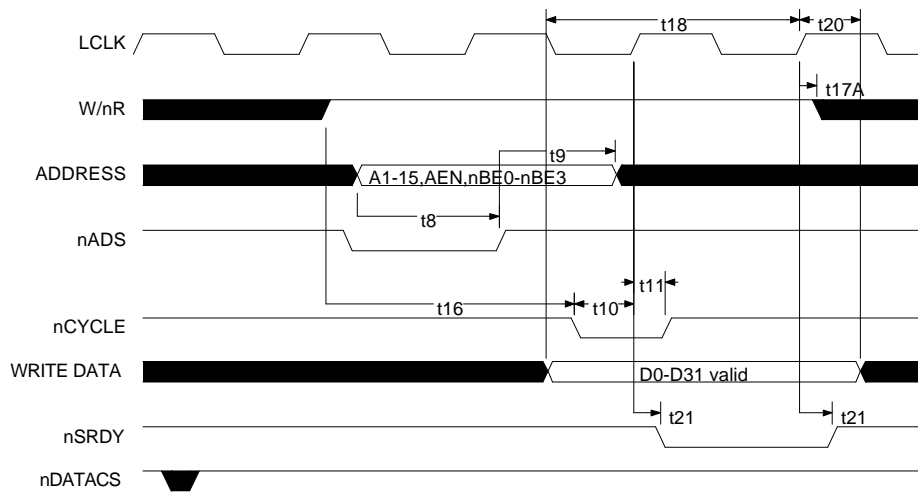
**FIGURE 18 - ADDRESS LATCHING FOR ALL MODES**



	PARAMETER	MIN	TYP	MAX	UNITS
t8	A1-A15, AEN, nBE0-nBE3 Setup to nADS Rising	10			ns
t9	A1-A15, AEN, nBE0-nBE3 Hold After nADS Rising	15			ns
t25	A4-A15, AEN to nLDEV Delay			20	ns



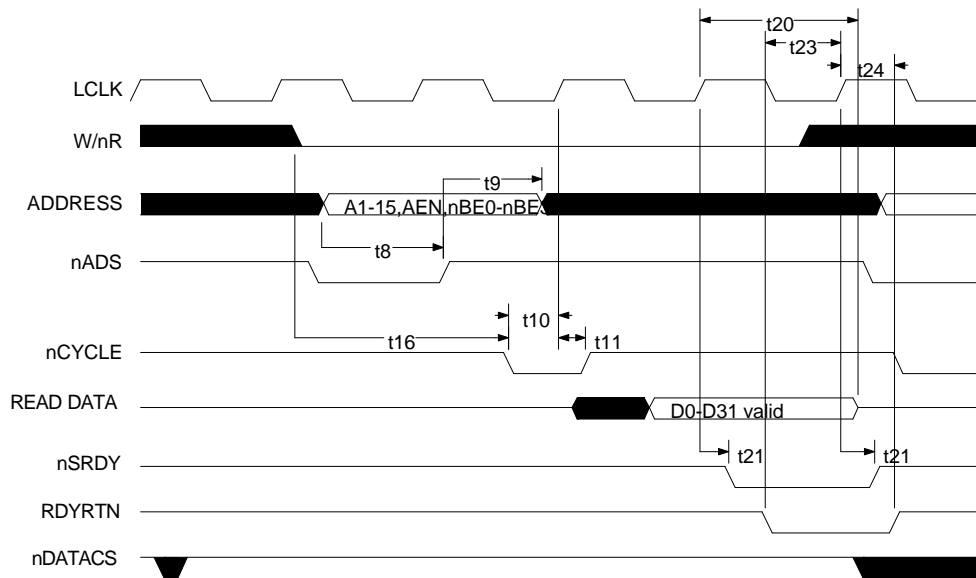
**FIGURE 19 - SYNCHRONOUS WRITE CYCLE - nVLBUS = 0**



	PARAMETER	MIN	TYP	MAX	UNITS
t8	A1-A15, AEN, nBE0-nBE3 Setup to nADS Rising	10			ns
t9	A1-A15, AEN, nBE0-nBE3 Hold After nADS Rising	15			ns
t10	nCYCLE Setup to LCLK Rising	7			ns

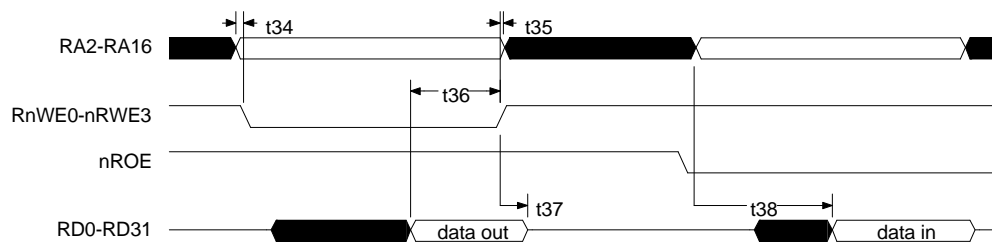
t11	nCYCLE Hold after LCLK Rising (Non-Burst Mode)	3			ns
t16	W/nR Setup to nCYCLE Active	30			ns
t17A	W/nR Hold after LCLK Rising with nLRDY Active	5			ns
t18	Data Setup to LCLK Rising (Write)	13			ns
t20	Data Hold from LCLK Rising (Write)	5			ns
t21	nLRDY Delay from LCLK Rising			10	ns

**FIGURE 20 - SYNCHRONOUS READ CYCLE - nVLBUS = 0**

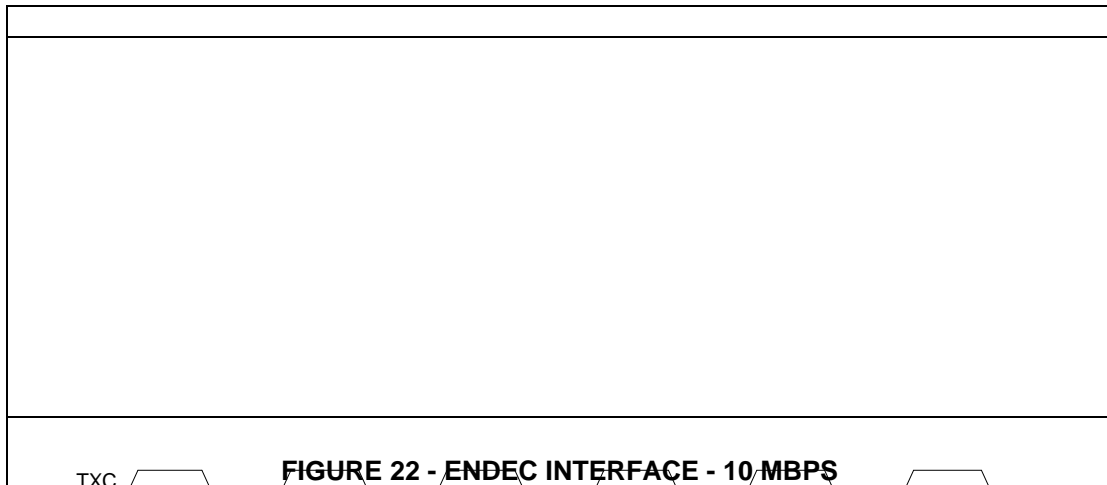


	<b>PARAMETER</b>	<b>MIN</b>	<b>TYP</b>	<b>MAX</b>	<b>UNITS</b>
t8	A1-A15, AEN, nBE0-nBE3 Setup to nADS Rising	10			ns
t9	A1-A15, AEN, nBE0-nBE3 Hold After nADS Rising	15			ns
t10	nCYCLE Setup to LCLK Rising	7			ns
t11	nCYCLE Hold after LCLK Rising (Non-Burst Mode)	3			ns
t16	W/nR Setup to nCYCLE Active	30			ns
t20	Data Hold from LCLK Rising (Write)	5			ns
t21	nLRDY Delay from LCLK Rising			10	ns
t23	nRDYRTN Setup to LCLK Rising	7			ns
t24	nRDYRTN Hold after LCLK Rising	3			ns

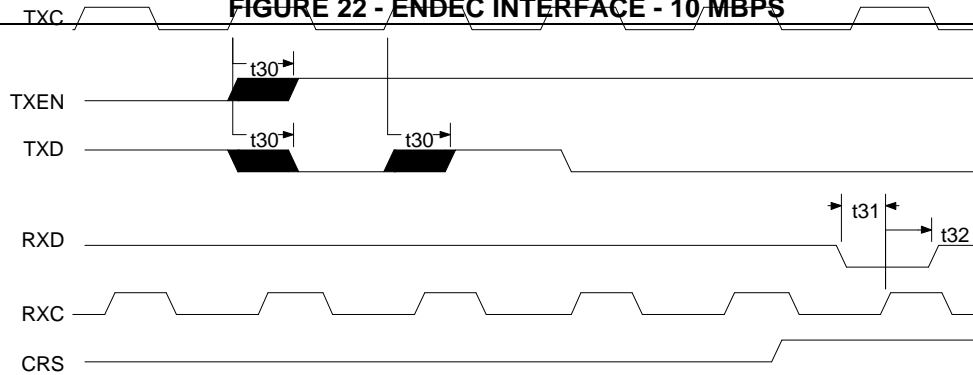
**FIGURE 21 - SRAM INTERFACE**



WRITE CYCLE		READ CYCLE			
	PARAMETER	MIN	TYP	MAX	UNITS
t34	RA2-RA16nn Setup to nRWE-0-nRWE3 Falling	0			ns
t35	RA2-RA16nn Hold After nRWE-0-nRWE3, nROE Rising	0			ns
t36	Write - RD0-RD31 Setup to nRWE0-nRWE3 Rising	12			ns
t37	Write - RD0-RD31 Hold after nRWE0-nRWE3 Rising	0			ns
t38	Read - RA2-RA16 Valid to RD0-RD31 Valid			25	ns



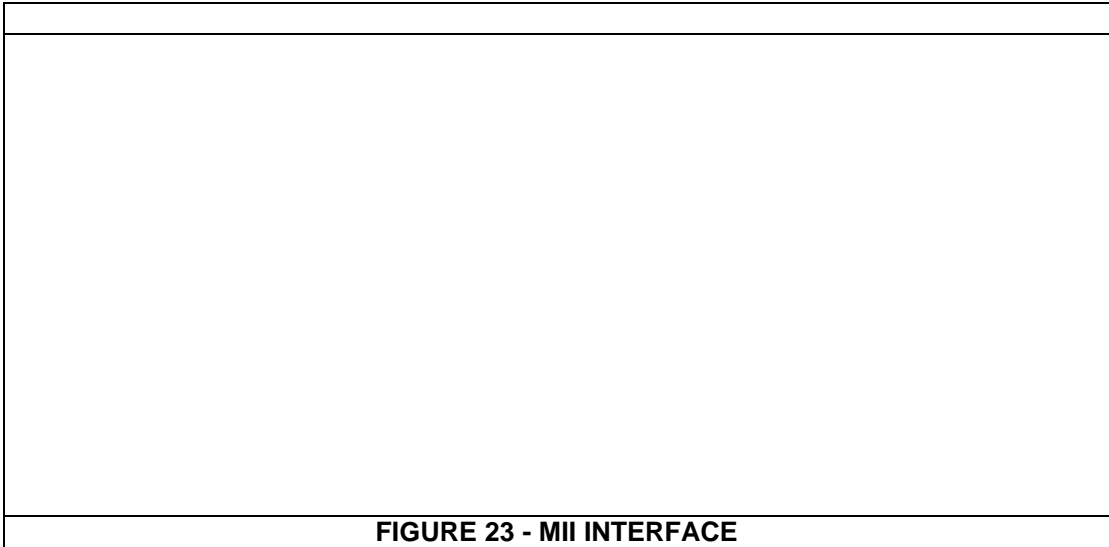
**FIGURE 22 - ENDEC INTERFACE - 10 MBPS**



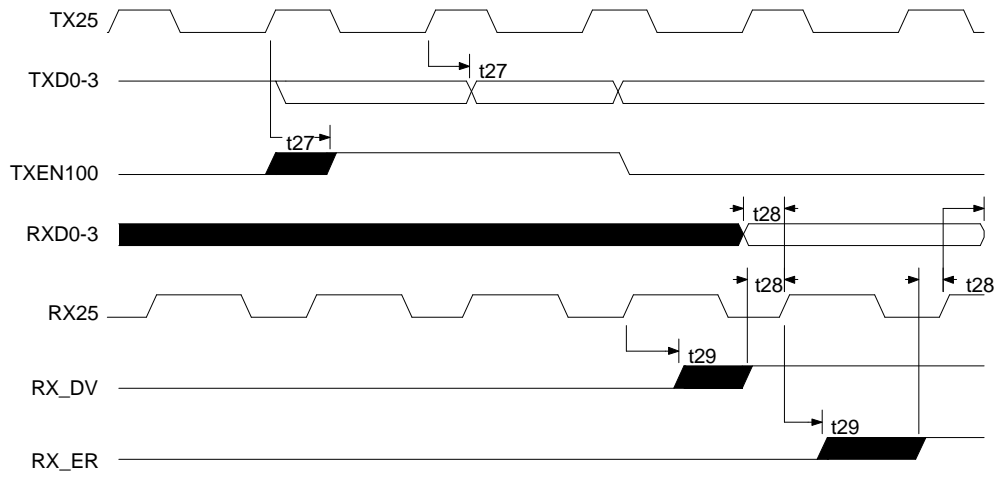
	PARAMETER	MIN	TYP	MAX	UNITS
t30	TXD, TXEN Delay from TXC Rising	0		40	ns
t31	nRXD Setup to RXC Rising	10			ns
t32	RXD Hold After RXC Rising	30			ns

Notes:

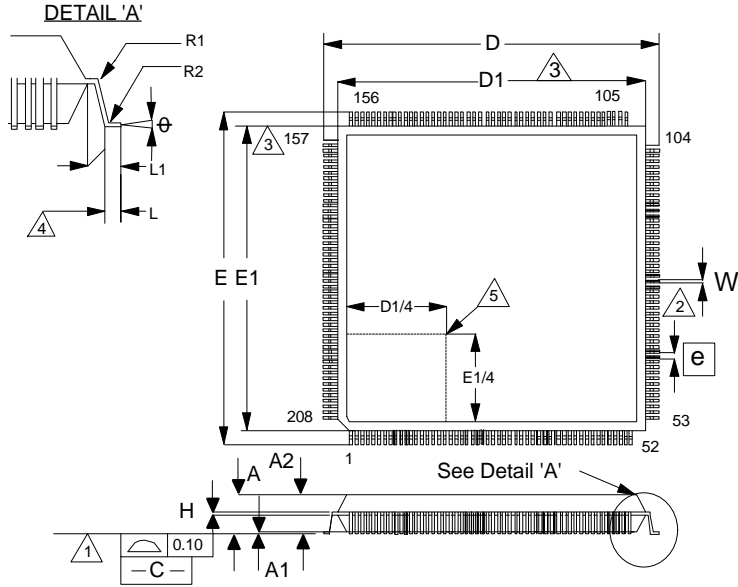
1. CRS input might be asynchronous to RXC.
2. RXC starts after CRS goes active. RXC stops after CRS goes inactive.
3. COL is an asynchronous input.



**FIGURE 23 - MII INTERFACE**



	PARAMETER	MIN	TYP	MAX	UNITS
t27	TXD0-TXD3, TXEN100 Delay from TX25 Rising	0		15	ns
t28	RXD0-RXD3, RX_DV, RX_ER Setup to RX25 Rising	10			ns
t29	RXD0-RXD3, RX_DV, RX_ER Hold After RX25 Rising	10			ns



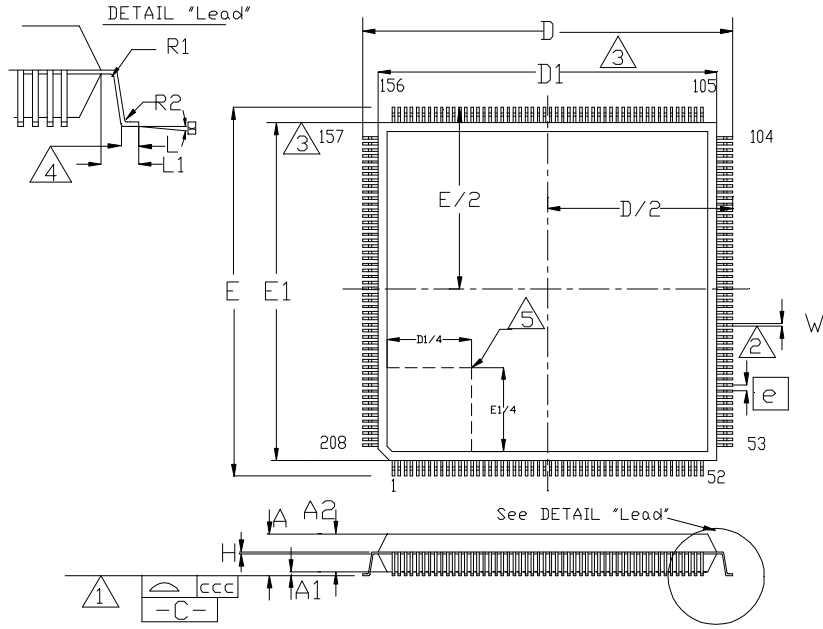
DIM	MIN	NOM	MAX
A			4.07
A1	0.05		0.5
A2	3.17		3.67
D	30.35	30.60	30.85
D1	27.90	28.00	28.10
E3	30.35	30.60	30.85
E1	27.90	28.00	28.10
H	0.09		0.23
L	0.35	0.5	0.65
L1		1.30	
e	0.50 BSC		
$\theta$	0°		7°
W	0.10	0.25	0.30
R1		0.20	
R2		0.20	

Notes:

- ① Coplanarity is 0.100mm maximum.
- ② Tolerance on the position of the leads is 0.08mm maximum.
- ③ Package body dimensions D1 and E1 do not include the mold protrusion. Maximum mold protrusion is 0.25mm.
- ④ Dimension for foot length L when measured at the centerline of the leads are given in the table. Dimension for foot length L when measured at the gauge plane 0.25mm above the seating plane, is 0.6mm.
- ⑤ Details of pin 1 identifier are optional but must be located within the zone indicated.
- 6. Controlling dimension: millimeter



**FIGURE 24 - 208 PIN PQFP PACKAGE OUTLINES**



DIM	MIN	NOM	MAX	REMARK
A			1.60	Overall Package Height
A1	0.05		0.15	Standoff
A2	1.35		1.45	Body Thickness
D	29.80	30.00	30.20	X Span
D/2	14.90	15.00	15.10	1/2 X Span Measure From Centerline
D1	27.90	28.00	28.10	X Body Size
E	29.80	30.00	30.20	Y Span
E/2	14.90	15.00	15.10	1/2 Y Span Measure From Centerline
E1	27.90	28.00	28.10	Y Body Size
H	0.09		0.23	Lead Frame Thickness
L	0.45	0.60	0.75	Lead Foot Length From Centerline
L1		1.00		Lead Length
e	0.50 BSC			Lead Pitch
$\phi$	$\phi$		$7^\circ$	Lead Foot Angle
W	0.17		0.27	Lead Width
R1	0.08			Lead Shoulder Radius
R2	0.08		0.20	Lead Foot Radius
ccc			0.0762	Coplanarity (Assemblers)
ccc			0.08	Coplanarity (Test House)

Notes:

- △ Controlling Unit: Millimeter.
- △ Tolerance on the position of the leads is  $\pm 0.04\text{mm}$  maximum.
- △ Package body dimensions D1 and E1 do not include the mold protrusion. Maximum mold protrusion is 0.25mm.
- △ Dimension for foot length L measured at the gauge plane 0.25mm above the seating plane, is 0.78-1.08mm.
- △ Details of pin 1 identifier are optional but must be located within the zone indicated.

**FIGURE 25 – 208 PIN TQFP PACKAGE OUTLINES**



©1997 STANDARD MICROSYSTEMS  
CORP.

**sm<sub>s</sub>c™**

Circuit diagrams utilizing SMSC products are included as a means of illustrating typical applications; consequently complete information sufficient for construction purposes is not necessarily given. The information has been carefully checked and is believed to be entirely reliable. However, no responsibility is assumed for inaccuracies. Furthermore, such information does not convey to the purchaser of the semiconductor devices described any licenses under the patent rights of SMSC or others. SMSC reserves the right to make changes at any time in order to improve design and supply the best product possible. SMSC products are not designed, intended, authorized or warranted for use in any life support or other application where product failure could cause or contribute to personal injury or severe property damage. Any and all such uses without prior written approval of an Officer of SMSC and further testing and/or modification will be fully at the risk of the customer.

LAN91C100 Rev. 9/24/97