VT86C926

AMAZON
PCI ETHERNET CONTROLLER

DATA SHEET
(Preliminary)

DATE : June 27, 1995

VIA TECHNOLOGIES, INC.

PRELIMINARY RELEASE
Please contact Via Technologies for the latest documentation.

Copyright Notice:

Disclaimer Notice:

No license is granted, implied or otherwise, under any patent or patent rights of Via Technologies. Via Technologies makes no warranties, implied or otherwise, in regard to this document and to the products described in this document. The information provided by this document is believed to be accurate and reliable to the publication date of this document. However, Via Technologies assumes no responsibility for any errors in this document. Furthermore, Via Technologies assumes no responsibility for the use or misuse of the information in this document and for any patent infringements that may arise from the use of this document. The information and product specifications within this document are subject to change at any time, without notice and without obligation to notify any person of such change.

Offices:

5020 Brandin Court
Fremont, CA 94538
USA

Tel: (510) 683-3300
Fax: (510) 683-3301

8th Floor, No. 533
Chung-Cheng Rd., Hsin-Tien
Taipei, Taiwan ROC

Tel: (886-2) 218-5452
Fax: (886-2) 218-5453

VT86C926 AMAZON
PCI ETHERNET CONTROLLER FEATURES

* Single chip Ethernet controller for PCI bus interface
* Software compatible to NE2000+, NE2000
* Support 35ns access time SRAM
* Integrated 10BaseT TP interface
  -- Built-in pre-equalization circuitry for transmitter
  -- smart squelch circuit with programmable threshold for receiver
  -- Selective Link integrity test and link disable
  -- Selective polarity detection and correction
  -- Automatic selection between TP and AUI interfaces
* Full IEEE 802.3 AUI interface
  -- Meet 10Base5 and 10Base2 standards
  -- Thin-net or Thick-net enable signal
* Jumper or Jumperless configuration
  -- Jumperl2ess configuration with EEPROM
  -- Jumper configuration with EEPROM storing Ethernet Address
* Direct drive to 4 status LED indicators: transmit, receive, collision, and Thin select
  -- Transmit LED for packet transmitting status and power indicator
  -- Receive LED for packet receiving status and Link status
  -- Collision LED for collision status
  -- Thin select LED for Thin 10Base-2 port select
* ISA NE2000 compatible mode, register compatible to National Semiconductor DP83905 & DP83906
* Optional Full duplex operation with speed up to 20Mbps
* Early receive interrupt allows parallel processing of data during receiving
* Early transmit correction ability allows parallel processing of data during transmission
* Support 16K, 32K Bootrom size
* Support PCI 32-bit data path transfaer with enhance 32-bit driver
* Software controllable power down feature
* Single +5V supply, 0.8um standard CMOS technology
* 100 pin PQFP package

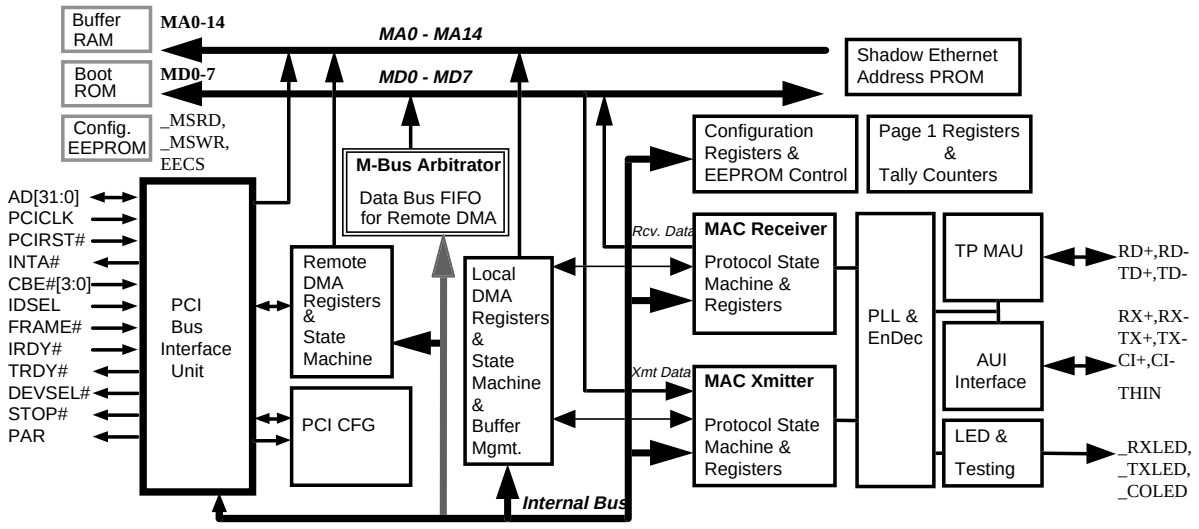


Figure 1: Block Diagram

PIN DIAGRAM

PIN DESCRIPTIONS

No.	Name	Type	Description
PCI Bus Interface			
98-100, 1-5, 8-13, 16-17, 26-33, 35-40, 43,44	AD31-0	I/O	Address/Data are multiplexed on the same PCI pins. A bus transaction consists of an address phase followed by one or more data phases. The address phase is the clock cycle in which FRAME# is asserted. Write data is stable and valid when IRDYB is asserted and read data is stable and valid when TRDYB is asserted.
95	PCICLK	I	PCICLK provides timing for all transactions on PCI and is an input pin to every PCI device.
96	INTA#	OD	INTA# is an asynchronous signal which is used to request an interrupt
97	PCIRST#	I	When PCIRST# is asserted low, the VT86C926 chip performs an internal system hardware reset. PCIRST# may be asynchronous to CLK when asserted or deasserted. It is recommended that the deassertion be synchronous to guarantee clean and bounce free edge.
6, 18, 25, 34	CBE#[3:0]	I	Bus Command/Byte Enables are multiplexed on the same PCI pins. During the address phase of a transaction, CBE3-0B define the Bus Command. Buring the data phase, CBE3-0B are used as Byte Enables. The Byte Enables define which physical byte lanes carry meaningful data. CBE0B applies to byte 0 and CBE3B applies to byte 3.
7	IDSEL	I	Used as a chip select during PCI configuration cycle.
19	FRAME#	I	Cycle Frame is driven by the current master to indicate the beginning and duration of an access. FRAME# is asserted to indicate a bus transaction is beginning. While FRAME# is asserted, data transfers continue. When FRAME# is deasserted, the transaction is in the final data phase.
20	IRDY#	I	Initiator Ready indicates the initiating agent's ability to complete the current data phase of the transaction. IRDY# is used in conjunction with TRDY#. A data phase is completed on any clock when both IRDY# and TRDY# are asserted. During a write, IRDY# indicates that valid data is present on AD31-0. During a read, it indicates the master is prepared to accept data. Wait cycles are inserted until both IRDY# and TRDY# are asserted simultaneously.
21	TRDY	I/O	Target Ready indicates the target's agent's ability to complete the current data phase of the transaction. TRDY# is used in conjunction with IRDY#. A data phase is completed on any clock when both IRDY# and TRDY# are asserted. During a read, TRDY# indicates that valid data is present on AD31-0. During a write, it indicates the target is prepared to accept data. Wait cycles are inserted until both IRDY# and TRDY# are asserted simultaneously.
22	DEVSEL#	I/O	Device Select, when actively driven, indicates the driving device has decoded its address as the target of the current access. As an input, DEVSEL# indicates whether any device on the bus has been selected.
23	STOP#	I/O	VT86C926 drives STOP# to disconnect further traction.
24	PAR	T/S	Parity is even parity across AD31-0 and CBE3-0B. PARis stable and valid one clock after the address phase. For data phaases PAR is stable and valid one clock after either IRDY# is asserted on a write transaction or TRDY# is asserted on a read transaction.

Network Interface			
85,86	TX+ TX-	O	**AUI Transmit Output:** Differential current mode driver which sends the Manchester-encoded data to the AUI DO circuitry through transformer coupler. Use a local terminator resistor 75 Ohm.
83,84	RX+ RX-	I	**AUI Receive Input:** Differential receive input pair form the transceiver through a decoupled transformer.
81,82	CD+ CD-	I	**AUI Collision Input:** Differential collision (SQE) signal input from transceiver. The SQE signal is 10MHz +/- 15% square wave.
91,92	TD+ TD-	O	**Twister Pair Transmit Outputs:** Different current mode driver which sends pre-equalized TP data signal to TP network medium through a filter and transformer. Use a local terminator resistor 100 Ohm.
89,90	RD+ RD-	I	**Twisted Pair Receive Inputs:** This differential input pair from TP medium through a decoupled transformer passes Manchester-encoded data to the ENDEC module.
80	THIN	O	**Thin-net selection:** This output is used to control the power DC-DC converter of 10Base2 MAU module. It is asserted when PHYS1 and PHYS0 in Configuration Register B selects 10Base2 mode.
69	X1	I	Crystal or external oscillator input.
68	X2	O	Cystal feedback output. Used in crystal connections only. Should be left completely unconnected when using oscillator module.
78	TXLED#	O	**Transmit/Power LED:** An open drain active low output. Normally on. It indicates a transmission onto the TP is in progress. It turns off for 90ms if a packet is transmitted. A minimum on time 18ms is maintained between blinking of transmitting packets.
79	RXLED#	O	**Receive/Link LED:** An open drain active low output. Normally on. It indicates a reception from the TP is in progress. It turns off 90ms if a packet is received. If the Link Test is enabled it will turn off as long as the link fails. After the link is up, it will turn on for at least 786ms and then become a normal receiver indicator.
77	COLED#	O	**Collision/Jabber LED:** An open drain active low output. Normally off. It turns on for minimum 25ms whenever VT86C926 detects a collision. It is retriggerable. During jabber state it stays on.
76	TSTPIN	I/O	N.C.

External Memory Support			
75-72, 67-58, 55	MA0-MA13, MA14	O	**Memory Support Address Bus:** When RESET is inactive these pins drive the memory support address bus. These 15 address lines cover up to 16K bytes memory space. When RESET is active these pins are used for jumper setting. There are 100K pull-down device on pins.
52-45	MD0-MD7	I/O	**Memory Support Data Bus:** When RESET is inactive these pins can be used to access external memory. When RESET is active pin MD3 to MD7 are used for jumper setting. There are 100K pull-down device on pins. MD0 to MD2 are used for EEPROM programming interface.
57	MSRD#	O	**Memory Support Bus Read:** Strobes data from external RAM into VT86C926 controller via the memory support data bus. (Note 1) **Note 1:** The Chip Select of external buffer RAM is always enabled.
56	MSWR#	O	**Memory Support Bus Write:** Strobes data from VT86C926 controller into the external RAM via the memory support data bus. A 100K pull-down device is connected.
54	BPCS#	O	**Boot PROM Chip Select:** Select the Boot ROM on the memory support data bus. The ROM address bus is connected to SA bus directly.
53	EECS	O	**EEPROM Chip Select:** Chip select signal for the external EEPROM when a EEPROM is used to provide the configuration data and Ethernet Address. A 100K pull-up resistor is connected.
Power Supply & Ground			
14, 42, 93, 71	VDD, VDDIF, VDD	P	**Positive 5V Supply:** Supply power to Internal digital logic, Digital I/O pads, and TD, TX pads. Double bonding may be required.
15, 41, 94, 70	VSS, VSSIF, VSS	G	**Negative Supply:** digital ground. Multiple bonding pads are required to separate core and I/O pads ground.
88	VDDA	P	**Analog 5V Supply:** Supply power to Analog circuit and RX, CI, RD pads. Seperate power plane may be required.
87	VSSA	G	**Negative Supply:** Analog ground. Sepearte ground plane may be required.

FUNCTIONAL DESCRIPTIONS

1. GENERAL DESCIPTION

The VT86C926 ***PCI*** Ethernet controller is CMOS VLSI designed for easy implementation of CSMA/CD local area networks. Significant features include: twisted-pair interface, PCI Plug&Play compatability and Early Interrupt Receive/Transmit.

The VT86C926 integrates the entire bus interface of PCI systems. Setting hardware jumpers or software configures the VT86C926 bus interface. The VT86C926 also complies with PCI specification v2.0. and is software compatible to either the NE2000. The VT86C926 supports the 10BASE5 or 10BAE2 network interfaces via an external transciever connected to its AUI port. The VT82C926's integrated 10BASE-T transciever supports 10BASE-T and fully complies with IEEE 802.3i specifications.

2. TP (TWISTED PAIR) INTERFACE

The five main functions of the TP interface are: smart squelch, collision detection, link detector/generator, jabber and transmit driver.

2.1. Smart Squelch

The basic function of smart squelch is to distinguish between impulse noise and valid signals on the RXI± differential inputs. To determine the validity of data, the squelch function applies specific timing and amplitute criteria to the incoming signals. Also, there are two squelch modes, selectable by the SQSEL pin. The two selectable squelch modes are: 1) 10BASE-T compatible and 2) reduced squelch mode.

2.2. Collsion

The TP interface will detect a collision when the transmit and receive channels are active at the same time. If the TPI module is receiving when a collision is detected it is reported to the controller immediately . Approximately 1 us after the transmission of each packet, a signal called the Signal Quality Error (SQE) is generated. The SQE consists typically of 10 cycles of 10 MHz and is also refered as the "Heartbeat."

2.3. Link Detector/Generator

The link generator is a timer circuit that generates a link pulse as defined by the 10 Base-T specification. The link pulse will be generated by transmitter. The pulse is used to check the integrity of the connection to the remote MAU. The link detection circuit checks for valid pulses from the remote MAU; if it receives no valid pulse, then the transmitter will be disabled.

2.4. Jabber

The jabber timer monitors the transmitter and disables the transmission if the transmitter is active for longer than 26 ms.

6. EEPROM INTERFACE AND PROGRAMMING

VT86C926 uses an 93C46 to store configuration data and Ethernet address.

6.1. EEPROM Contents

	D15	D0
3FH	Reserved for 93C46	Reserved for 93C46
.	.	.
.	.	.
.	.	.
.	.	.
.	.	.
.	.	.
10H	.	.
0FH	**73H**	Conf. C
0EH	Conf. B	Conf. A
	. Reserved .	. Reserved .
08H	57H	42H
07H	SUBVID1	SUBVID0
	. Reserved .	. Reserved .
03H	Checksum	Board ID
02H	Ethernet Address 5	Ethernet Address 4
01H	Ethernet Address 3	Ethernet Address 2
00H	Ethernet Address 1	Ethernet Address 0

Note 1. The word on location 03H is optional to user's application requirement.
Note 2. Programming 73H into the upper address is required to protect the Ethernet address from being destroyed accidentally.

6.2. Emulated PROM map for NE2000 16-bit mode (WTS=1)

	D15	D0
1EH	00	**57H** (DWID=1) or **42H** (DWID=0)
1CH	00	**57H** (DWID=1) or **42H** (DWID=0)
	. 00 .	. Reserved .
0EH	00	Checksum
0CH	00	Board ID
0AH	00	Ethernet Address 5
08H	00	Ethernet Address 4
06H	00	Ethernet Address 3
04H	00	Ethernet Address 2
02H	00	Ethernet Address 1
00H	00	Ethernet Address 0

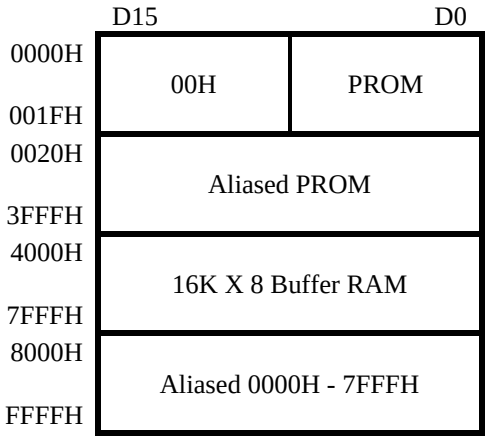
Note 1. The address shown above on the left side is the Remote DMA Address.
Note 2. The jumper DWID will determine the contents of the upper bytes. The software uses the upper bytes to distinguish the data width of the system bus interface to be used.
Note 3. During the initialization of the software driver, the program should set WTS to 1 and try 8-bit I/O instruction to get the PROM contents (since data width is unknown). Then the program should set the WTS bit of DCR to the value revealed on

jumper DWID. Remember that as long as WTS is set to 1, the VT86C926 sends one word at one time with _IOCS16 asserted -- regardless of the data width of I/O instruction used.

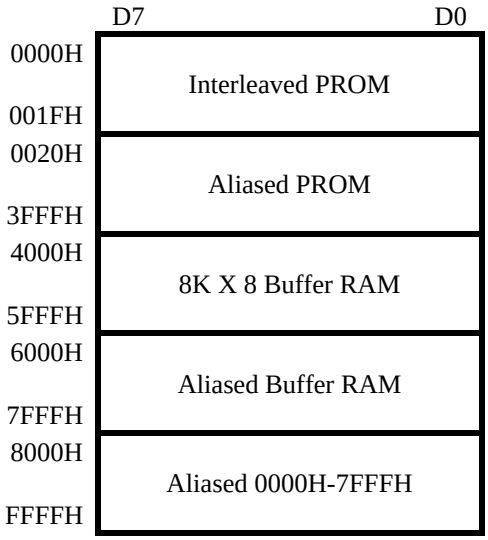
Note 4. Regardless of the data width of I/O instruction used, the VT86C926 sends one byte at one time with _IOCS16 deasserted and WTS set to 0. This map is only for reference. Users should never need to access the PROM contents in this way.

6.3. Memory Map of VT86C926

NE2000 16-bit mode (WTS=1)



NE2000 8-bit mode (WTS=0)



6.4. PCI Configuration Space

Device ID (0926)				Vendor ID (1106)								00 h
STATUS (DEVS1, DEVS0) = (1 , 0)				**COMMAND** **(MMSPACE, IOSOACE)**								04 h
CLASS CODE (02_00_00)								**Revision ID** **(00)**				08 h
BIST (00)		Header type (00)		Latency Timer (00)		Cache Line (00)						0c h
IO SPACE							0	0	0	0	1	10 h
Reserved				*Reserved*								
				SUB Vendor ID FR EEPROM contents								2c h
EXP ROM BASE [31: 15]				ROM14	0000_0000_00000			EN				30 h
Reserved				*Reserved*								
Max_LAT (00)		Min_GNT (00)		INT PIN (01)		INTLINE INTL [7:0]						3c h

6.5. Direct Programming of EEPROM

The VT86C926 features a easy way to program external EEPROM in-situ. When the RESET is active and if the upper byte of 0FH on EEPROM is not 73H, the EEPR bit will not be set to indicate that the current EEPROM has not been programmed yet. This will allow the VT86C926 to enter Direct Programming mode if EELOAD is also set. In this mode the user can directly control the EEPROM interface signals by writing to the ECSR Port and the value on the EECS, ESK and EDI bits will be driven onto the EECS, SK(MD2), and DI(MD1) outputs respectively. These outputs will be latched so the user can generate a clock on SK by repetitively writing 1 then 0 to the appropriate bit. This can be used to generate the EEPROM signals as per the 93C46 data sheet.

To read the EEPROM data, users have to generate EEPROM interface signals into EECS, DI and SK as described above and in the mean time read the data from DO(MD0) input via pin SD0. Reading Data Transfer Port during programming will not affect the latched data on EECS, SK, and DI outputs. When the EEPROM has been programmed and verified (remember to program the upper byte of 0EH with 73H), the user must give VT86C926 a power-on reset to return to normal operation and to read in the new data.

The Direct Programming mode is mainly used for production to program every bit of the EEPROM. Once the upper byte of 0EH has been programmed with 073H and a power-on reset has been performed, there is

no way to change the contents of EEPROM except Configuration Registers A, B, and C, which will be discussed in the following paragraph. For more information, refer to EECSR.

6.6. Embedded Programming of EEPROM

If the upper byte of 0FH of EEPROM has been programmed to 073H when VT86C926 is loading the EEPROM data during power-on reset, the EEPR bit of Signature Register will be set to prohibit the Direct Programming mode. However, the user can still program the configuration registers A, B, and C using the Embedded Programming mode by following the routine specified in the pseudo code below. This operation will work regardless of the value of EECONFIG. The setting of the EELOAD bit of Configuration Register B starts the EEPROM write process. Care should be taken not to accidentally modify the POL and GDLNK bits because these two bits return the value indifferent from the setting. This programming process is ended when the EELOAD bit goes to zero.

```
EEPROM_EMB_PROG ( )
{    // defined constant: CONFIG_B, GDLNK, POL, EELOAD
     // declared register: value, config_for_A, config_for_B, config_for_C
     // declared function: DISABLE_INTERRUPTS, ENABLE_INTERRUPTS, READ, WRITE, WAIT
     DISABLE_INTERRUPTS ( );
     value = READ (CONFIG_B);
     value = value & !GDLNK & !POL;
     value = value | EELOAD;
     WRITE (CONFIG_B, value);
     READ (CONFIG_B);
     WRITE (CONFIG_B, config_for_A);
     WRITE (CONFIG_B, config_for_B);
     WRITE (CONFIG_B, config_for_C);
     while (value || EELOAD)
     {
          value = READ (CONFIG_B);
          WAIT ( );
     }
     ENABLE_INTERRUPTS ( );
}
```

7. CONTROL AND STATUS REGISTERS

* VT86C926 supports the control and status registers of DP8390 except those explained as follows.
* VT86C926 supports all page 1 registers. Only part of Page 2 is supported.
* VT86C926 supports Early Transmit Underrun (ETUN)
* VT86C926 supports most of page 0 registers.
* The meaning and use of 01H (CLAD0) and 02H (CLAD1) of page 0 is altered.
* The 06H (FIFO port) of page 0 is not supported.
* The following control/status bits in page 0 are not supported:
    - -- (D3,D4,D5) == (0,1,1) of CR (00H) : Send Packet Command (RD0 - RD2)
    - -- D1 of DCR: Byte Order Select (BOS)
    - -- D2 of DCR: Long Address Select (LAS)
    - -- D4 of DCR: Auto-initialize Remote (ARM)
    - -- D5, D6 of DCR: FIFO Threshold Select (FT0 and FT1)
    - -- D4 of TCR: Collision Offset Enable (OFST)
    - -- D5 of TSR: FIFO Underrun (FU)
    - -- D7 of TSR: Out of Window Collision (OWC)
    - -- D3 of RSR: FIFO Overrun (FO)

8. POWER DOWN MODE

The VT86C926 provides an one level power down mode. The BIOS or Network OS device driver can configure Register A to diagnostic mode then set the Power-on bit of the diagnostic port to "on." When the VT86C926 is in Power down mode, all power to the analog port is cut off and the chip clock is stopped. Other registers are read only. Only the diagnostic port is read-writeable.

9. JUMPERED OPERATION SUPPORT

9.1 Fully Jumpered Operation

Figure 8 below depicts this operation. In this configuration, most options are selected by jumpers on the VT86C926 controller memory bus

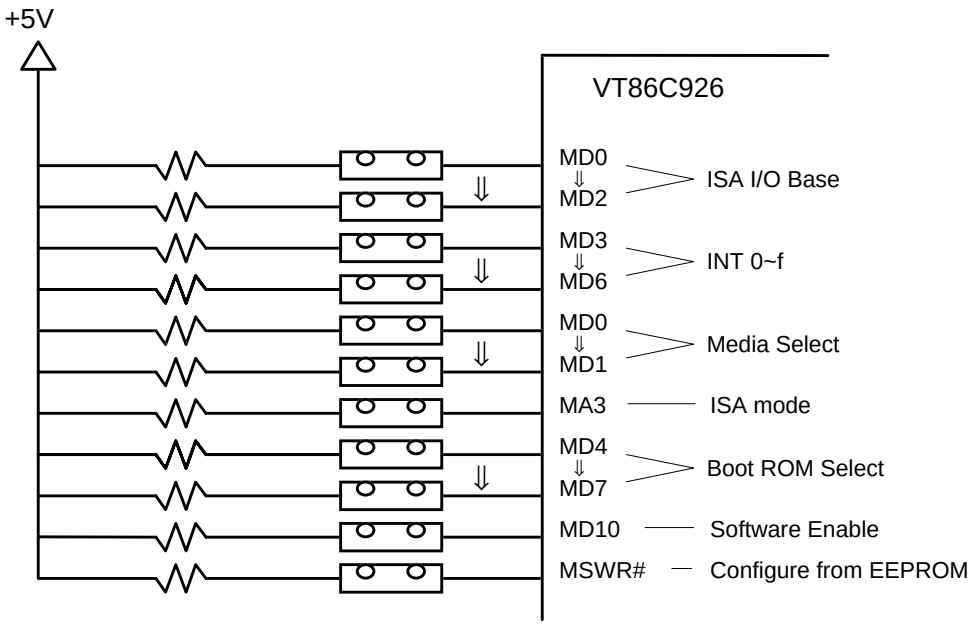


Figure 3: Example of Jumper Configuration

10. PACKET TRANSMISSION DESCRIPTION

The VT86C926 NIC is CSMA/CD compliant and offers scheduled retransmission of packets up to 15 times on collisions according to the truncated binary exponential backoff algorithm. The following is an overview of the transmission process.

10.1.Transmission Setup

In order for the transmit packet to be ready for transmission, it must follow the IEEE 820.3 specification (see figure 4). When the transmit packet is set up in the memory, the VT86C926 will initialize the packet starting address (TPSR) and the packet length (TPCR0, TBCR1). Then the PTX bit (transmit packet) of the command register will be set to start transmission.

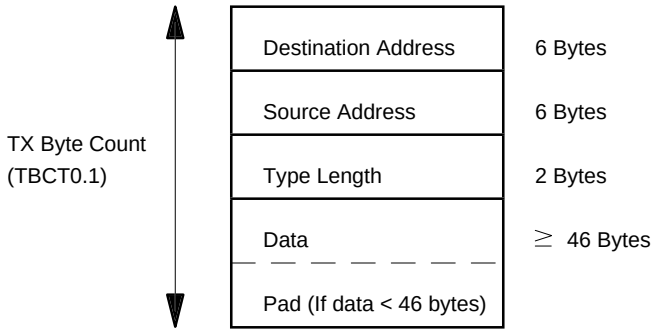


Figure 4. Transmit Packet Format

10.2.Transmission Process

After the PTX bit is set to start transmission, the TXE (transmit enable) signal is asserted and transmission begins (see figure 5). The CRC is calculated after the 62 bit preamble (alternating ONEs and ZEROs) and the start of the frame delimiter (two ONEs) are sent out as NRZ data (pin TxD) with a clock (TxC). This process does not end until the byte count reaches zero. After all bytes are serialized, the 4 CRC bytes are then serialized and added to the packet. In case of collisions, transmissions would cease and jam sequence of 32 ONEs would be transmitted to notify every node of the collision. Once the jam sequence is transmitted, the packet would be rescheduled for transmission according to the truncated binary exponential backoff algorithm.

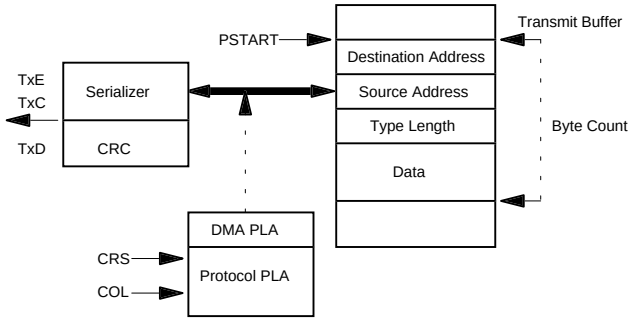


Figure 5. Packet Transmission

10.3.Transmission Status

An interrupt is generated at the end of transmission: either the PTX bit (complete packet transmitted) or the TXE bit (packet transmission aborted) of the ISR (Interrupt Status Register) is set. This interrupt routine discerns the specific details of the transmission by reading the TSR. If the PTX bit is set, reading the TSR can reveal the following: if a carrier was present when the transmission was initiated (DFR), if the carrier was lost during the transmission (CRS -- This would point to a short somewhere on the network), if the collision detect circuitry is working properly (CDH) and if a collision occured (COL). The collision count register (NCR) is incremented each time a collision occurs. The out of window collision (OWC) bit of the TSR is set whenever a collision occurs outside the 512K window (slot time). If 16 collisions occur, then the TXE but if the ISR is set. If 16 collisions abort a transmission, then the ABT bit of the TSR is set.

11. PACKET RECEPTION

The VT86C926 uses a receive filter scheme to manage packet reception. Because every node receives all packet transmissions in CSMA/CD compliant networks, this receive filter is needed to filter out all packets without matching addresses.

11.1.Reception Process

The VT86C926 will begin to check for the two consecutive ONEs than indicates the start of the frame delimiter (SFD) once VT86C926 dectects the active CTS signal and the ONE-ZERO preamble. After the detection of the SFD, the incoming data is deserialized and sent to the address filter (see figure 6). If the packet passes the address filter, then the packet is sent to the receive buffer ring through the DMA. An interrupt is generated when the packet is received into the buffer ring.

To determine the acceptance and rejection of packets, the calculated value of the CRC, at the point in which the CRS signal is low, is compared with the last four bytes received. If a match occurs on the last byte boundary, then the packet is accepted. If the CRCs do not match at the last byte boundary and the CRC goes LOW, the packet is rejected and an CRC error is indicated (CRC bit of RSR set); that is to say the recieve buffer ring is not updated. If the CRCs do not match at the last byte boundary and the CRC does not go LOW, then a frame alignment error occurs and is flagged (FAE bit of RSR set). However, frame alignment errors occur only when CRC errors occur.

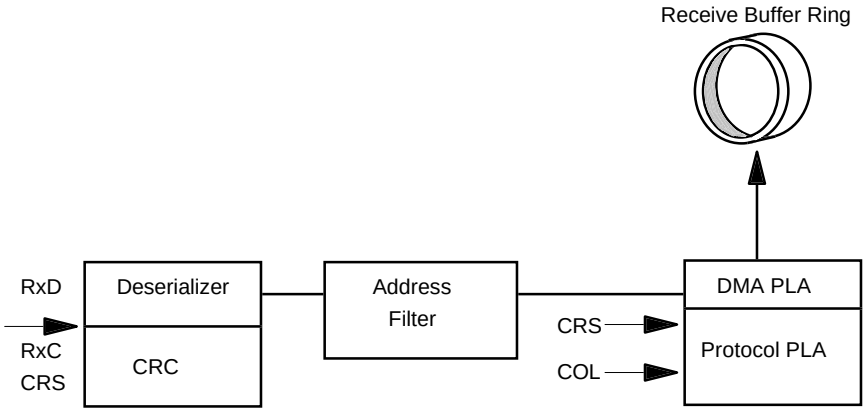


Figure 6. Packet Reception

11.2.Receive Buffer Ring

Incoming packets are placed by the NIC into the Receive Buffer Ring to await processing. As each packet is processed, it is then removed from the ring. The NIC allocates an area of memory to use as the buffer ring during initialization. Because ring pointers are located on the chip, the buffer management scheme works at a high efficiency. The DMA channels can transfer data at 10 Mbyte per second. The second DMA or the remote DMA channel removes packets from the rings after the packets have been processed. Though the receive buffer ring allows packets to be processed in the order of arrival, it is not a FIFO buffer. It is just a portion of memory that works effectively as a FIFO buffer

The four on-chip pointers that control the ring are the page start (PSTART) pointer, page stop (PSTOP) pointer, current page (CURR) pointer and boundary (BNRY) pointer. The PSTART and PSTOP pointers determine the size of the buffer ring. The CURR pointer determines the location of the next packet to be received and the BNRY pointer determines the location of the next packet to be removed. When packets are received, the CURR pointer moves to the next location (in the direction of rotation) and the BNRY follows the CURR pointer. However, the PSTART and PSTOP remains the same throughout the procedure.

To receive packets, buffers are linked as necessary to accomodate the incoming packets. Up to 256 buffers can be linked together in a single ring. Each buffer is 256 bytes. That effectively yields a maximum of 64K bytes for the entire ring. Therefore, the theoretical maximum space between PSTART and PSTOP is 64K and all the pointers refer to 256 byte boundaries (see figure 7).

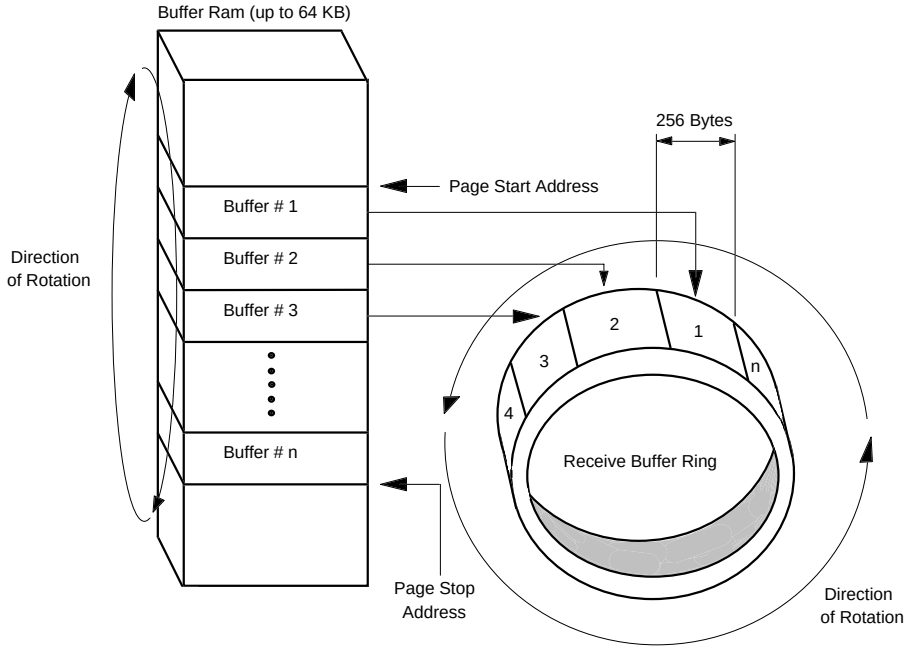


Figure 7. Receive Buffer Ring

The pointers PSTART and PSTOP are determined during initialization. The first valid packet received is placed at the CURR pointer plus a four byte offset (see figure 8). If the packet is greater than the 256 byte buffer, then the buffers are linked together until the packet is received in its entirety. Because of the ring design of the buffer, the first and the last buffer may be linked together. When a packet is completely received, the status from the Receive Status Register (RSR), a pointer to the next packet and the byte count of the current packet are written into the four byte offset.

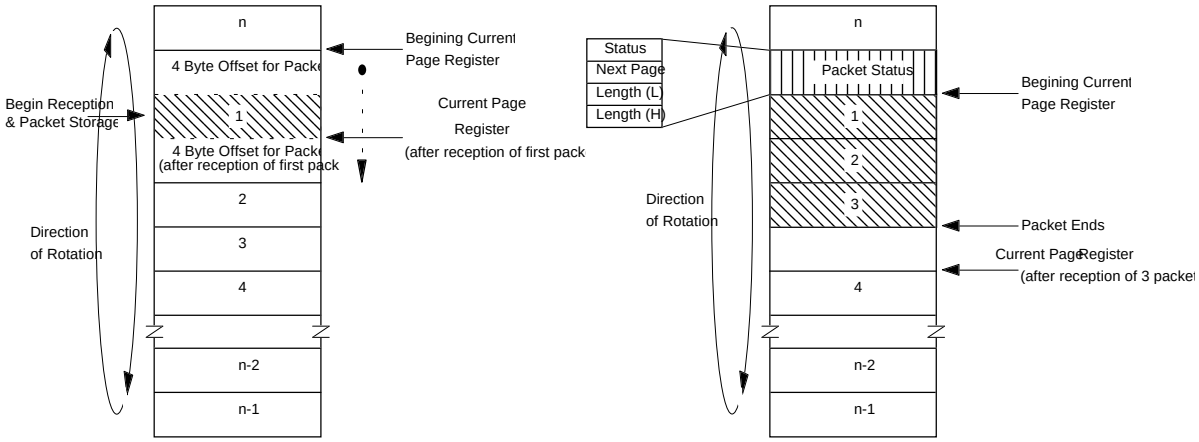


Figure 8. Receive Packet Buffering

Note: Figures 5, 6, 7 and 8 are flat illustrations of the Receive Buffer Ring. See figure 4 for graphical conceptualization.

When an error occurs, be it FAE or CRC, it is automatically overwritten. The CURR is not moved to next buffer boundary, hence the next incoming packet will overwrite the previous packet (see figure 9). The error-overwrite feature can be toggled on and off by setting the save errored packet (SEP) bit in the RCR. Runt packets are also overwritten. The runt packet overwrite may be toggled on and off by setting the AR bit in the TCR.

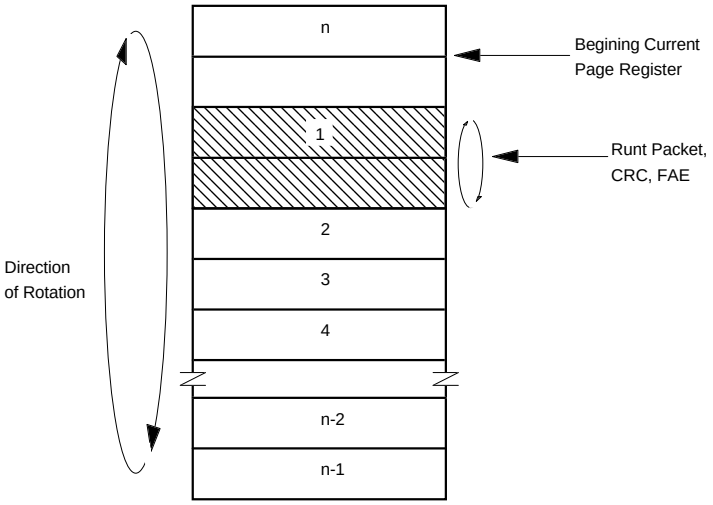


Figure 9. Packet Rejection

The BNRY is updated each time a packet is removed. In that way, the BNRY always follow the CURR around the ring (see figure 10). However, in situations in which the packet processing is slower than the packet reception, the ring may be full. When the ring becomes full, all incoming packets are aborted so that packets already in the ring would not be overwritten (see figure 11). New packets will start to arrive after the BNRY is updated. The missed packet tally counter would keep track of all missed packets. This situation is unlikely if enough memory is allocated for the receive buffer ring. If the BNRY is located outside the buffering, the overwrite protection function will not be available. All incoming packets would overwrite packets already on the ring, whether the packets on the ring have been processed or not.

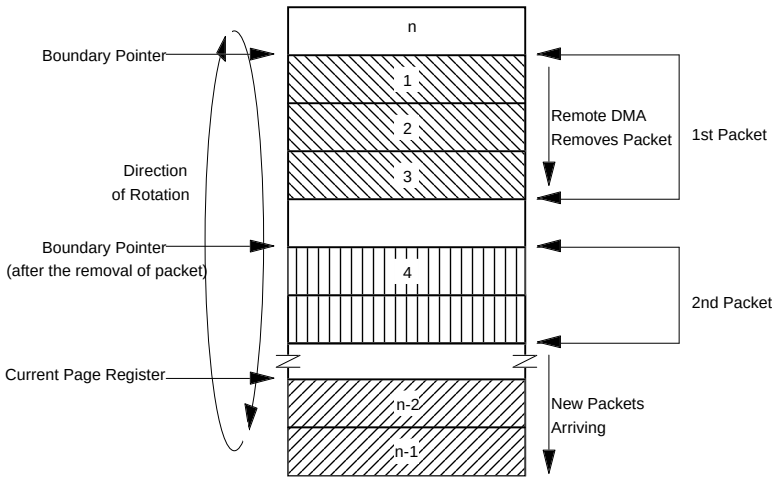


Figure 10. Removing Packets from the Receive Buffer Ring

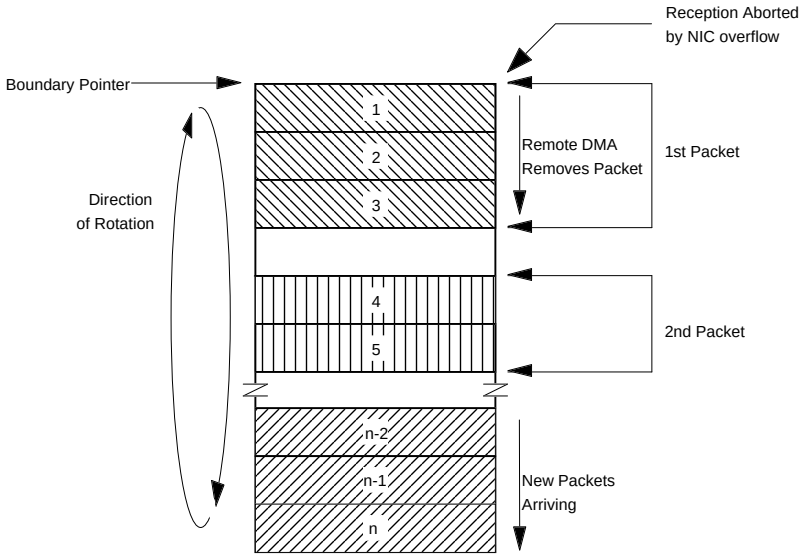


Figure 11. Receive Buffer Ring Overwrite Protection

12. EARLY INTERRUPT RECEIVE MODE

The VT86C926, under normal mode, will generate an interrupt to the host after a complete packet is received and decide to accept it. This, however, will introduce a throughput latency at least longer than the transmission time of the received packet. To allow the host driver program look ahead the contents of the packet under receiving, the VT86C926 provides the Early Interrupt Receive Mode which is enabled by setting the EIEN bit of Diagnostic Register.

During the Early Interrupt Receive Mode, several interrupts are generated based on the receiving conditions as depicted in the following paragraph.

1. Packet header interrupt
   When an incoming packet matches the address (physical or not) of the network controller and the total received bytes count exceeds 64, the VT86C926 will generate an interrupt to notify the host an addressed packet in on the way. The driver can distinguish this early interrupt by reading ISR and RSR. If the ISR shows an PRX status and RSR doesn't reflect that status on its PRX bit, an early interrupt is being generated.

2. Packet page interrupt
   Every time the receiver buffer is completly filled and a new buffer is requested, the VT86C926 will generate an interrupt with PRX of RSR unset as Packet header interrupt. Therefore, this kind of interrupt is issued every 256 byte time (~200us) until packet reception is done. The current byte count of packet being received can be accessed from register 01H (CLDC0) and 02H (CLDC1) of page 0.

3. Receive status interrupt
   This is the normal interrupt generated after an addressed packet is completely received. If the received packet is a good packet, both the ISR and RSR will set PRX bit high. Otherwise, the RXE or OVW will be reflected on ISR and FAE, XRXE, or MPA on RSR.

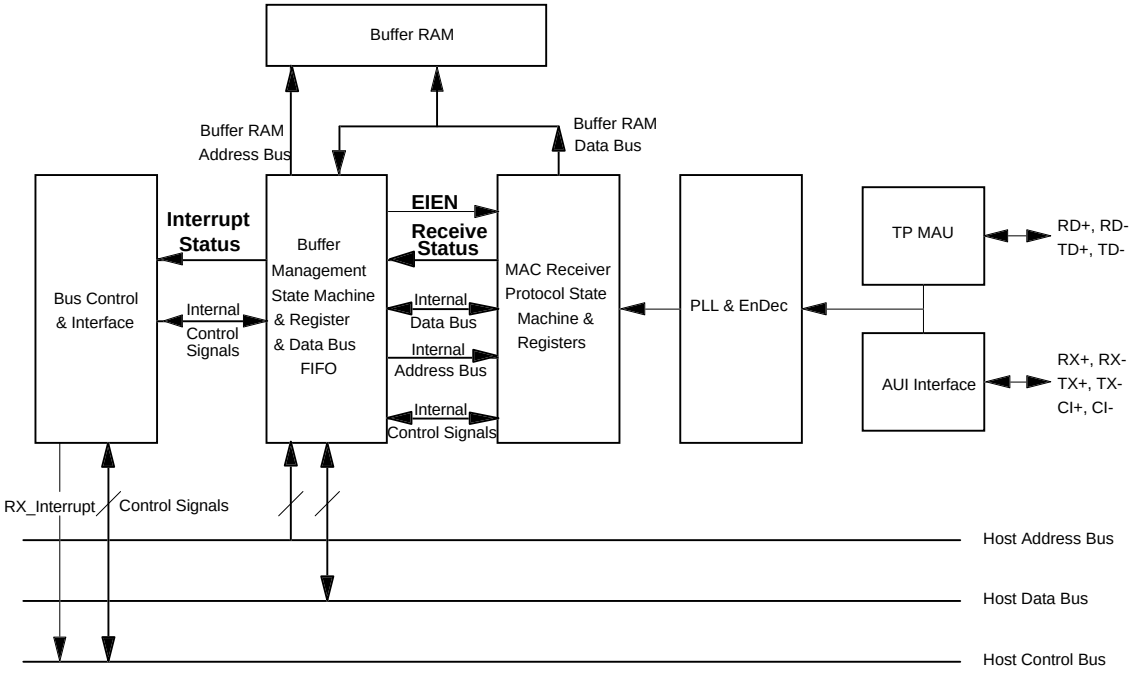


Figure 12: Early Interrupt Receive Block Diagram

12.1. Early Interrupt Architecture Programming Algorithm

12.1.1. Enable Early Interrupt Mode

To set the VT86C926 to Early Interrupt mode, set bit 3 (EIEN) of the Diagnostic Register to high. To set the VT86C926 back to normal mode, set bit 3 (EIEN) of the Diagnostic Register to low. You can set Early interrupt mode in any state. But your receive interrupt service routine must able to receive fragments from Ethernet data. If the routine cannot handle the fragment data, you may get corrupt data.

12.1.2. Early interrupt Receive Routine Refence register

The receive interrupt service routine algorithm fetches one page data from the SRAM upon reception of an interrupt. Refer to the following register and status bit for Early Interrupt service routine:

Interrupt Status Register (ISR, 0x07(0) bit0 PRX, bit 2 RXEE)
Receive Status Register (RSR,0x0C(0) bit0 EARLY PRX)
Current Local DMA Counter Low and High (CLDC(0), 0x01, 0x02)
Current Page Register (CURR, 0x07(1))

But we suggest that you use this method as below, in order to reduce the interrupt occured times. Please refer the CLDC counter for currently received bytes.

This algorithm sample enables the VT86C926's early interrupt drivers to increase the LAN system peer to peer transfer rate. Programmers can modify the orignal NE2000 driver sources to add this function.

1. When driver is initialized,
   set PktPointer = 0 and allocate PktBuffer = 1.5 KByte.

2. When interrupt is received, check ISR, CURR and RSR.

```
If RXE bit in ISR has been set, reset PktPointer = 0. (it is a packet error interrupt, STATE 0)
Else If PRX bit in ISR has been set, compare CURR with BNRY+1,

    If CURR == BNRY+1, (it is a early interrupt condition)
        Read and save CLDC (Current Local Data Count)
        Evenlize CLDC. (CLDC = CLDC - CLDC % 2)

        If PRX bit in RSR has not been set,
            If RXE bit in ISR has been set, goto packet error condition (STATE 0).
            If CLDC <= PktPointer, goto packet error condition (STATE 0).
            RSAR = ((BNRY + 1) << 8) + PktPointer.
            If RSAR >= PSTOP, set RSAR = RSAR - (PSTOP - PSTART).
            RBCR = CLDC - PktPointer.
            Use remote DMA to read packet data into offset PktBuffer + PktPointer.
            PktPointer = CLDC.

        Else goto packet complete condition (STATE 1).

    Else (it is a packet complete interrupt, STATE 1)
        RSAR = (BNRY + 1) << 8.
        RBCR = 4.
        Use remote DMA to packet header into offset PktBuffer.
        If RXE bit in ISR has been set, set PktPointer = 0.
        Get PktLen = packet length from PktBuffer.
        PktLen = (PktLen + 1) - (PktLen + 1) % 2.
        if PktLen <= PktPointer, set PktPointer = 0.
```

RSAR = ((BNRY + 1) << 8) + PktPointer.
RBCR = PktLen - PktPointer.
Use remote DMA to read packet data into offset PktBuffer + PktPointer.
PktPointer = 0.

If PktPointer = 0,
Do the normal driver packet received procedure,
but get packet data from PktBuffer, not the receive buffer ring.
Else go to interrupt entry to poll ISR again.

Rx condition	ISR	RSR	CURR	CLDC
Packet header or page data received	1	0	BNRY+1	<>0
Total packet received	1	1	BNRY+1 + (packet pages)	= 0
More then one packet	1	0	BNRY+1 + (packet pages)	<>0
CRC or page align error	4	4	X	X

13. EARLY TRANSMIT MODE

In order to provide the VT86C926 with parallel processing between the ISA bus and Ethernet media, the VT86C926 must be set to detect early transmit error mode. This error may occur when the system's master device uses the ISA bus and then uses the data bus.

If the ETEN bit is set in the configuration register B, the transmit status status bit5 (ETUN) and the transmit error status bit (TXE) in the interrupt service register (ISR) bit 3 would be set on. If the interrupt mask register TXEE bit is set to on, an interrupt will be generated when the local DMA pass the remote DMA machine.

This mode is essentially similar to the Early Interrupt Receive Mode. When the VT86C926 is under the normal mode, the CPU can issue the "transmit packet" command to the VT86C926 and the internal local DMA will tranfer the data from the SRAM to the Ethernet. However, the packet data must already exist in SRAM.

In the situation in which the "transmit packet" command is issued to the local DMA and the remote DMA (to fill SRAM) at the same time, a corrupt packet may be transmitted to the network and remain undetected.

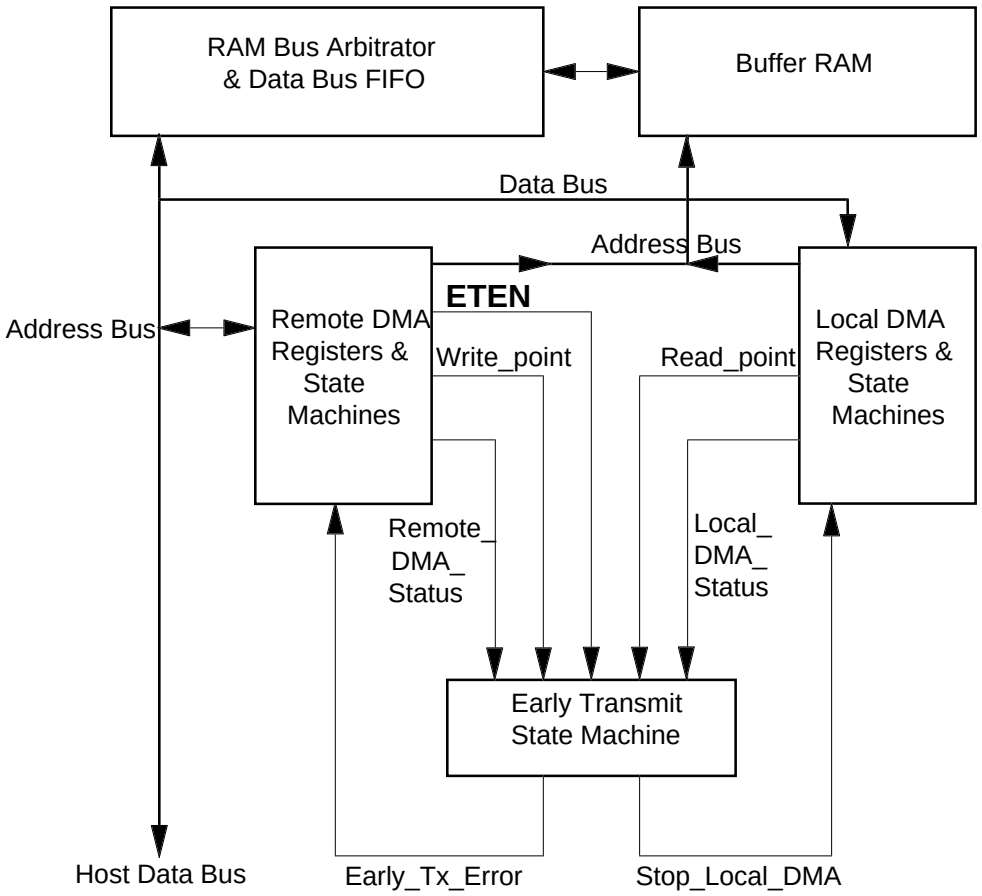


Figure 13: Early Transmit Error Detect Block Diagram

Note: If Write_point<Read_point, then Early_Tx_Error.
If Write_point>Read_point, then Stop_Local_DMA.

PROJECT SIGN OFF SHEET

FOR VT86C926 PRELIMINARY RELEASE

	NAME	SIGNATURE	DATE
EDITOR	Alex Tsao		
PRODUCT SPECIFICATIONS	Jeffery Chang		
PROJECT DESIGNER	Antonio Weng		
FINAL APPROVAL	Tzu-Mu Lin		