

# RIC™-SONIC™ Interface

National Semiconductor  
Application Note 782  
Imad Ayoub  
April 1993



RIC-SONIC Interface

## INTRODUCTION

This document describes how the DP83950 Repeater Interface Controller (RIC) can be interfaced to a System Oriented Network Interface Controller (SONIC) controller. The emphasis in this note is on the hardware interface between the RIC and the SONIC. The software implementation of the Hub management protocols such as SNMP and CMIP are not discussed in this note, since each system's implementation would be different depending upon the processor used and the number of RICs and SONICs employed in the Hub. A description of the extra logic necessary to interface the RIC to a NIC (DP8390) is included for reference. And last, in order to provide a simple and fast solution for evaluating the RIC's management bus interface to the SONIC, a description of a simple way to hook the SONIC DP839EB-ATS evaluation board to the RIC's management Bus is included.

## RIC-SONIC INTERFACE

The RIC transmits over the management bus every packet that is received from any of the ports (refer to the RIC datasheet for details). The management bus packet is different from the packets transmitted to/from the ports. First, the preamble on this bus is always five bits (01011). Second, at the end of the packet, after the CRC pattern, seven bytes of management status are appended to the packet by the RIC. These seven bytes are always aligned to start on a byte boundary. Third, the packet is in NRZ format.

A properly connected and configured SONIC receives every packet that is sent over the management bus, and therefore buffers the data as well as the seven bytes of status. The Packet Compression feature available on the RIC and the SONIC allows for specific handling of the data part of the packet, as described later.

Figure 1 shows the interface of one RIC to one SONIC. The SONIC is configured to run in the external decoder mode to receive the NRZ data from the management bus (refer to the SONIC datasheet for more details). The SONIC input pins CRS, RXC, and RXD tie directly to the RIC management bus output pins MCRS, MRXC and MRXD (with the RIC BINV selected for active high signals, refer to the RIC datasheet for details). The SONIC runs in Promiscuous Mode accepting all the packets from the management bus.

The packet compression output pin  $\overline{PCOMP}$  of the SONIC ties directly to the  $\overline{PCOMP}$  input pin of the RIC. The SONIC can be programmed to assert  $\overline{PCOMP}$  upon a match or a mismatch of the packet destination address with a SONIC CAM address. For managed Hub applications, the SONIC asserts  $\overline{PCOMP}$  if the destination address of the received packet does not match any address in the CAM of the SONIC. For managed bridge applications the SONIC asserts  $\overline{PCOMP}$  if the destination address of the received packet matches any address in the CAM of the SONIC.

The managed Hub application is selected in this implementation. If the packet is addressed to the SONIC, the  $\overline{PCOMP}$  pin will not be asserted by the SONIC and the RIC will not compress the data. The SONIC receives the whole packet with the seven bytes of status. If the packet is not addressed to the SONIC, the  $\overline{PCOMP}$  pin will be asserted by the SONIC. The RIC will compress the data by inhibiting the clocks during the data part of the packet, and will re-enable the clock during the seven bytes of status.

The SONIC buffers the seven bytes of management status from the RIC to memory. These bytes can then be accessed by a processor. Utilizing the packet compression technique leads to an implementation that minimizes memory requirements, i.e., buffering only the data needed by the SONIC and the seven bytes of status. The RIC contains a Packet Compress Decode Register that can be used to determine the number of bytes, post SFD, which are transferred over the management bus when the packet compression option is employed.

Since the seven bytes of status are appended after the CRC pattern, the SONIC will indicate that a CRC error occurs every time a packet is received. This should be ignored, and the SONIC should be set to save errored packets. The CRC bit in the seven bytes of status appended to the packet will indicate whether the packet has a CRC error or not.

To enable the SONIC to transmit to the network, the SONIC of Figure 1 transmits a packet to the RIC through the Inter-RIC bus. The SONIC transmit signals TXE, TXD tie to the Inter-RIC signals IRE and IRD through a TRI-STATE® buffer (74F125), which is TRI-STATE when the SONIC is not transmitting. Since the SONIC is in external decoder mode, the TXC pin is an input. An external 10 MHz oscillator provides the input to the TXC pin of the SONIC and the IRC pin of the RIC. The SONIC will drive ACTN and ACKI of the RIC as soon as it wants to transmit. Driving ACTN informs the RIC that the SONIC wants to transmit. In this implementation the SONIC is placed on top of the arbitration chain with the RIC, therefore the SONIC drives the ACKI input of the RIC when it wants to transmit.

The management bus does not experience any collisions, however any collisions on the network detected by the RIC are reported in the seven bytes of status. There will be no receive collisions on the SONIC, and the SONIC does not drive any collision signals to the RIC. The SONIC needs to be notified when a transmit collision occurs on the RIC. Therefore the COL input pin of the SONIC is driven by the ANYXNd output of the RIC whenever there is a transmit collision on the RIC and the SONIC is transmitting.

TRI-STATE® is a registered trademark of National Semiconductor Corporation.  
RIC™ and SONIC™ are trademarks of National Semiconductor Corporation.

AN-782

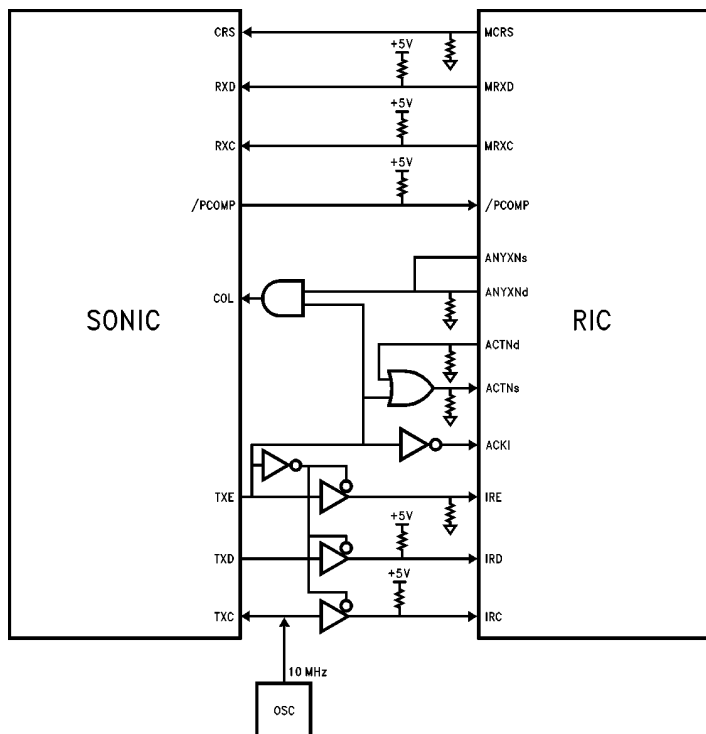


FIGURE 1

TL/F/11231-1

Figure 2 shows an implementation with several RICs sharing one SONIC for a managed Hub application. The SONIC is set in Promiscuous Mode to receive all packets, and it asserts  $\overline{PCOMP}$  upon address mismatch. The Hub is addressable, and the SONIC can receive and transmit packets to the network as well as receive the seven bytes of RIC management status. All RICs share a single management bus to send the data to the SONIC. The SONIC transmits through the Inter-RIC bus to all RICs.

To interface the SONIC to the RIC in this implementation, a PAL is needed to generate the following signals:

$$\begin{aligned} \overline{ACKO} &= \overline{ACKI} \& TXE + \overline{ACKI} \\ ACTNd &= \overline{ACKI} \& TXE \\ ANYXNd &= TXE \& \overline{ACKI} \\ COL &= TXE \& ANYXNs \\ TXEO &= TXE \& \overline{ACKI} \end{aligned}$$

This implementation utilizes the serial arbitration method, and allows the SONIC to be placed anywhere in the arbitration chain.  $\overline{ACKO}$  is asserted if the  $\overline{ACKI}$  from the RIC above it is not asserted and the SONIC wants to transmit, i.e., TXE is asserted, or if  $\overline{ACKI}$  from the RIC above it is asserted. ACTNd is asserted to tell all RICs that it wants to transmit when ACKI is not asserted and TXE is asserted. The SONIC could experience a transmit collision in this implementation since it could be in the middle of the arbitration chain. ANYXNd is asserted by the SONIC when TXE is asserted, and  $\overline{ACKI}$  is asserted by any RIC higher in the arbitration chain. The SONIC is notified of a collision if it is transmitting and any RIC asserts ANYXN. Finally, TXEO is enabled

when the SONIC wants to transmit and  $\overline{ACKI}$  from the RIC above it is not asserted.

**RIC-NIC INTERFACE**

Any design that utilizes any controller other than the SONIC, such as the NIC (DP8390), to interface to the RIC should address the following points:

First, the packet compression feature of the RIC cannot be used by other controllers unless an external CAM and associated logic is used to generate the  $\overline{PCOMP}$  signal to the RIC. If this logic is available the controller may not operate properly with the clock inhibited. The SONIC has an on board CAM, and asserts  $\overline{PCOMP}$  to the RIC, and it works properly while the clocks are inhibited.

Second, due to the nature of the CSMA/CD protocol, there are situations when a collision will occur early in the packet (before SFD). This will lead to a packet transmitted onto the management bus containing only the seven bytes of status. This will be ignored by most controllers. Therefore extra logic will be required to stretch such packets to the controller's minimum acceptable packet length. The SONIC accepts such packets normally.

Third, knowing that the packet compression feature cannot be used, all packets that are transmitted over the network will need to be buffered by the controller. This requires a larger memory space, and may require a faster CPU.

Fourth, the SONIC will receive back to back packets from the management bus without missing packets due to insufficient gap (provided it is given access to memory). Other controllers may miss some packets if the gap is small.



**OTHER INTERFACE METHODS**

The method described so far to interface the SONIC to the RIC's management and Inter-RIC busses is not the only way to interface a controller to the RIC. A controller could also transmit and receive packets through the Inter-RIC bus or through any of the ports. However these two methods do not allow the controller to obtain the management bus data from the RIC. There are several drawbacks for not receiving this data:

First, even though part of the information available in the seven bytes of status is available through the CPU bus of the RIC, the CRC error flag, the Collision Bit Timer, the Repeat Byte Count, and the Inter Frame Gap Bit Timer are not available from the RIC except through the management bus.

Second, every packet transmitted through the management bus contains the number of the port receiving the packet. If the management bus is not used, the only way to obtain the port number is by setting the RIC to generate a Real Time Interrupt to the processor on every packet received. The processor then reads the Real Time Interrupt register to find out which port received this packet.

Third, in a multi-RIC system, the RIC number is essential for associating the packets with the receiving RIC and receiving port. This is included in the seven bytes of management status, and cannot be obtained otherwise directly from the RIC.

Fourth, the packet sent over the management bus contains the Source and Destination addresses, and the Packet Compress Decode Register can be used to specify the number of bytes to send over the management bus before inhibiting the clocks when  $\overline{PCOMP}$  is used. To perform this operation otherwise extra dedicated logic is needed to receive every packet on the network to read and save this data.

**SONIC EVALUATION BOARD MODIFICATION (DP839EB-ATS ONLY)**

This section describes a way to interface the SONIC to receive packets from the management bus of the RIC and to use the packet compression feature, using the Repeater

Evaluation Kit (RICKIT) and a DP839EB-ATS board. A new SONIC evaluation board, the DP83932EB-AT, will not require modification. Refer to AN-855 for more information. Contact your National Semiconductor representative regarding availability.

All that is needed for the SONIC to receive the management bus data is to tie CRS, RXO, RXD and  $\overline{PCOMP}$  pins from the SONIC directly to the MCERS, MRXC, MRXD and  $\overline{PCOMP}$  pins of the RIC (see *Figure 1*). This can be achieved as follows:

1. Place the DP839EB-ATS board in external decoder mode by removing the EXT jumper in the JB2 block, and removing all the jumpers in the JB4 block.
2. Take an SNI (DP8391, or DP83910) chip and clip off pins 2, 3, and 4, and place it in the appropriate socket (U18) on the DP839EB-ATS board.
3. Solder three wires to pins 2, 3, and 4 on the back of U18 on the DP839EB-ATS board and solder the other end to a female connector attached to the prototype area of the board.
4. To utilize the packet compression feature, use a SONIC (DP83932B) (pin 26 is the  $\overline{PCOMP}$  pin). Bend pin 26 up in order for it to be accessible after inserting the SONIC back into the socket. Solder one end of a fourth wire to this pin and solder the other end to the fourth pin of the connector on the prototype area.
5. On the RICKIT (DP83950EB-AT) Main Board, solder four wires to the pin side of R31, R71, R40 and R36. Connect these wires to a female connector. These four wires should be in the proper order to correspond to the proper pins from the DP839EB-ATS board.
6. Make a four wire ribbon cable that is approximately four inches long with a male pin connector at each end. This can now be used to connect between the two male connectors on the DP839EB-ATS board and the RICKIT Main Board.

The DP839EB-ATS board now has the proper modification to receive the management bus data from the RICKIT Main Board. These boards can now be installed into the same PC-AT using the diagnostic/evaluation software provided with the board. See the software manual provided with the board, for details.

**LIFE SUPPORT POLICY**

NATIONAL'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT OF NATIONAL SEMICONDUCTOR CORPORATION. As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and whose failure to perform, when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.
2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.



**National Semiconductor Corporation**  
 2900 Semiconductor Drive  
 P.O. Box 58090  
 Santa Clara, CA 95052-8090  
 Tel: 1(800) 272-9959  
 TWX: (910) 339-9240

**National Semiconductor GmbH**  
 Livry-Gargan-Str. 10  
 D-82256 Fürstenfeldbruck  
 Germany  
 Tel: (81-41) 35-0  
 Telex: 527849  
 Fax: (81-41) 35-1

**National Semiconductor Japan Ltd.**  
 Sumitomo Chemical  
 Engineering Center  
 Bldg, 7F  
 1-7-1, Nakase, Mihama-Ku  
 Chiba-City,  
 Chiba Prefecture 261  
 Tel: (043) 299-2300  
 Fax: (043) 299-2500

**National Semiconductor Hong Kong Ltd.**  
 13th Floor, Straight Block,  
 Ocean Centre, 5 Canton Rd.  
 Tsimshatsui, Kowloon  
 Hong Kong  
 Tel: (852) 2737-1600  
 Fax: (852) 2736-9960

**National Semicondutores Do Brazil Ltda.**  
 Rue Deputado Lacorda Franco  
 120-3A  
 Sao Paulo-SP  
 Brazil 05418-000  
 Tel: (55-11) 212-5066  
 Telex: 391-1131931 NSBR BR  
 Fax: (55-11) 212-1181

**National Semiconductor (Australia) Pty, Ltd.**  
 Building 16  
 Business Park Drive  
 Monash Business Park  
 Nottingham, Melbourne  
 Victoria 3168 Australia  
 Tel: (3) 558-9999  
 Fax: (3) 558-9998

National does not assume any responsibility for use of any circuitry described, no circuit patent licenses are implied and National reserves the right at any time without notice to change said circuitry and specifications.