DP83932 SONIC™ Bus Operations Guide

National Semiconductor
Application Note 745
Wesley Lee
July 1995

This application note is intended to be a supplementary document for the DP83932 SONIC datasheet, expanding upon the bus functional descriptions found in the datasheet. It is recommended that you are familiar with the bus operations of the SONIC before reading this document.

This application note gives additional examples of the SONIC's bus operations to illustrate a broader picture of receptions, transmissions, etc. Where possible, special conditions are included to show all conceivable bus operations performed by the SONIC. Detailed figures are shown to enhance clarity. This document is divided into two sections for bus master and slave operations. The bus master section details bus operations during transmission, receptions and load CAM operations, and the slave access section describes SONIC register accesses during idle and non-idle conditions.

TERMS AND ABBREVIATIONS

In this document certain terms and abbreviations will be used to describe the bus operations of the SONIC. These words are defined as follows:

Block Transfer:	A multiple transfer bus operation in which the address increments for each transfer.
Word:	Refers to a 16-bit quantity; a double word is a 32-bit quantity.
Memory Cycle:	The basic cycle which the SONIC reads from or writes to memory.
Bus Tenure:	The complete time the SONIC uses the bus during a block transfer.
Bus Latency:	This is the time from when the SONIC requests the bus to when the SONIC is granted the bus.
CAM:	Content Addressable Memory
TDA:	Transmit Descriptor Area
TBA:	Transmit Buffer Area
RRA:	Receive Resource Area
RDA:	Receive Descriptor Area
RBA:	Receive Buffer Area
CDA:	CAM Descriptor Area

1.0 BUS MASTER OPERATIONS

1.1 The Basic Block Transfer Cycle

The basic transfer cycle of the SONIC is composed of three basic operations: (1) acquiring the bus, (2) transferring data onto/from the bus, and (3) relinquishing the bus. Operations (1) and (3) are described in detail in Section 5.4 of the DP83932 datasheet or Section 7.3 of the DP83934 datasheet and will not be discussed here. Operation (2), however, will be more fully explained.

When the SONIC uses the bus, it transfers data to/from one specific area in memory (i.e., RBA, RDA, RRA, TDA, or TBA) as indicated by the bus status pins S<2:0>. If the SONIC needs to transfer the data to multiple areas in memory, it deasserts its bus request (HOLD or $\overline{\mathrm{BGACK}}$), then requests the bus again (HOLD or $\overline{\mathrm{BR}}$). During its tenure on the bus, the SONIC transfers a programmed number of words to memory, depending on where data is placed. The number of transfers to the descriptor areas (TDA, RDA, RRA, and CDA), are shown in the following table. Note that (during SONIC descriptor access) since the upper word (D<31:16>) is not used in 32-bit mode, the number of transfers are the same for both 16-bit and 32-bit modes.

TABLE 1-1. Number of Memory Transfers to the Descriptor Areas

Area	Number	R/W	When
CDA	4	R	All bus tenures except the last one
	5	R	Last bus tenure. The additional access is to load the CAM Enable register.
TDA	6	R	First descriptor fetch
	3	R	Additional fragment pointer and size fetches, if any
	2	R/W	Status and link access
RDA	7	R/W	Updating receive descriptor information
	6	R/W	Updating receive descriptor information but SONIC has read EOL = 1
	2	R/W	Re-reading RXpkt.link and writing to RXpkt.in_use when EOL has previously been detected as 1. The SONIC writes to the in_use field when EOL now reads 0.
	1	R	Re-reading the RXpkt.link as above but the SONIC still reads EOL = 1.
RRA	4	R	All bus tenures

RRD-B30M105/Printed in U. S. A.

DP83932 SONIC Bus Operations Guide

AN-745

For buffer area transfers (TBA and RBA), the number of memory transfers is determined by the FIFO threshold and whether the SONIC is in ‘‘empty/fill’ or ‘‘exact block’’ transfer modes, programmed in the Data Configuration register. For ‘‘exact block’’ transfer mode, the SONIC transfers the same number of words (or double words) as are programmed for the FIFO threshold. For example, if you programmed 4 double words as the threshold for the receive FIFO, the SONIC will transfer this amount of data to memory per bus tenure. There are two exceptions to this rule, however. First, during transmission or reception, if the packet is not a multiple of the FIFO threshold, the last bus tenure will contain less transfers than the FIFO threshold. Second, for high transmit FIFO thresholds (12 words or 14 words), the SONIC will fill the transmit FIFO only as much as needed to completely fill it (and not overfill it). Thus, if you choose a 12 word transmit FIFO threshold, the first bus tenure will transfer 12 words, but the second tenure will only transfer 4 words (12 words + 4 words = 32 bytes). This last example assumes that the bus latencies are zero.

For ‘‘empty/fill’’ mode, the number of transfers is also dependent on the bus latency. When the FIFO threshold has been reached, the SONIC will either completely empty the FIFO during reception or completely fill the FIFO during transmission. At the time of the bus request, the FIFO threshold equals the number of words in the FIFO, but due to bus latencies, additional bytes may have entered the FIFO (during reception). Thus, the number of words transferred during a bus tenure in this mode is the FIFO threshold plus the additional bytes that have entered the FIFO during reception or minus the bytes that have been serialized during transmission.

1.2 Packet Reception

This section gives a step-by-step description of the SONIC receiving a 68-byte packet. The initial conditions are shown below.

Initial conditions:

(a) The incoming packet is one that the SONIC will accept.

(b) The SONIC has detected that EOL = 1 from the previous reception, but the software has subsequently appended another receive descriptor before receiving this packet.

(c) The packet begins on a double word boundary.

(d) The Data Configuration register has been configured for:

- 32-bit data width mode (DB5 = DW = 1)
- 4 double word Receive FIFO threshold (DB3,2 = RF1,0 = 1,0)
- Exact Block Transfer mode (DB4 = BMS = 1)

(e) The packet has crossed over the End of Buffer Count (EOBC) register during this reception; hence the SONIC will need to use another Receive Buffer Area (RBA).

The reception is described as follows.

Note: The numbers in this section correspond to the numbers in *Figure 1-1*.

(1) Because of condition (a), the SONIC reads the RXpkt.link again to see if the software has subsequently reset the EOL bit to zero. Since it has (condition [b]), the SONIC writes to the RXpkt.in__use field and buffers the packet to the RBA. Note that this step is skipped if the SONIC has sufficient descriptors.

(2) Once the receive FIFO has reached its threshold (4 double words), the SONIC will write 4 double words during its bus tenure. For a 68-byte packet, the SONIC will perform this operation 4 times.

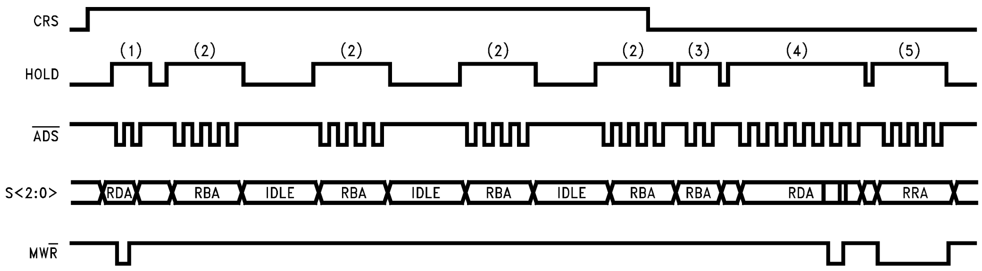
(3) During the last RBA bus tenure, the packet has ended (CRS goes low). The SONIC requests the bus once again (in 3 bus clocks) and flushes the remaining bytes in the FIFO (8 bytes). The first 4 bytes are the remainder of the packet and the last 4 bytes are the receive status that is automatically written into the FIFO by the SONIC. These last 4 bytes are extraneous to the RBA and are overwritten during the next reception. The usable receive status is written to the Receive Descriptor Area.

Note 1: If the packet size is not a multiple of the memory transfer size (16 bits or 32 bits), the SONIC will pad the last memory transfer with 1's as necessary.

Note 2: If any of the last 4 bytes exceeds the length of the Receive Buffer Area, these bytes will not be written to memory.

(4) The SONIC writes the status information in the Receive Descriptor Area. The SONIC performs 7 consecutive memory transfers during its bus tenure (5 writes to the RXpkt.status, RXpkt.byte__count, RXpkt.pkt__ptr0, RXpkt.pkt__ptr1, and RXpkt.seq__no. fields, 1 read to the RXpkt.link field, and 1 write to the RXpkt.in__ use field). See *Figure 1-3* and Section 1.2.2 for further details.

(5) Because of condition (e), the SONIC requests the bus again (in 3 bus clocks) and fetches a resource descriptor from the Receive Resource Area. The SONIC reads this area in 4 consecutive memory read operations.

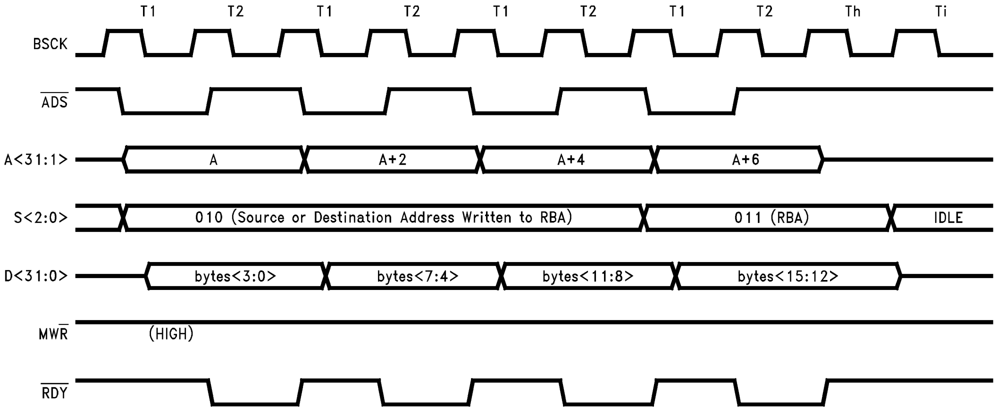


TL/F/11139–1

FIGURE 1-1. Complete Reception of a 68-Byte Packet

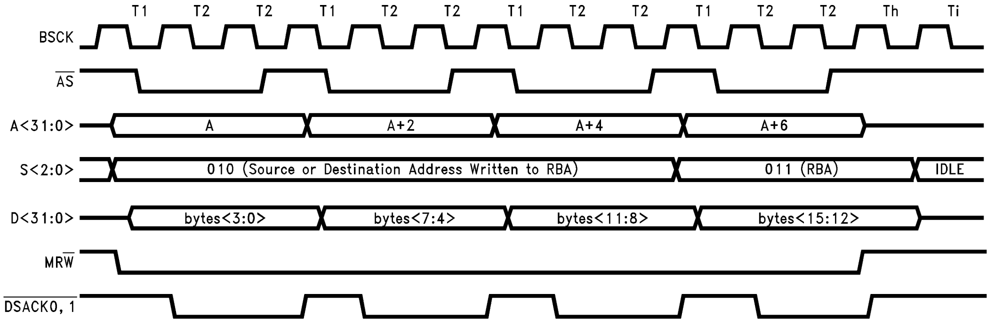
1.2.1 Detail of Access to the RBA

Figure 1-2a and *1-2b* show the first SONIC access to the RBA for BMODE = 0 and 1. Note that the status pins S<2:0> change from 0,1,0 to 0,1,1 as the SONIC finishes writing the Source Address of the packet and continues buffering the rest of the packet.



TL/F/11139–2

FIGURE 1-2a. First RBA Access for Storing Packet (BMODE = 0, Synchronous Mode)

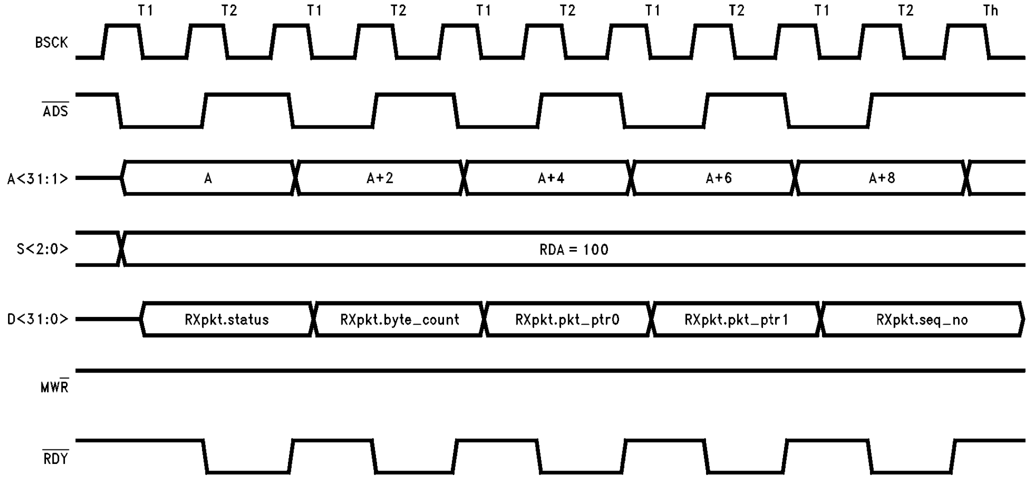


TL/F/11139–3

FIGURE 1-2b. First RBA Access for Storing Packet (BMODE = 1, Asynchronous Mode)

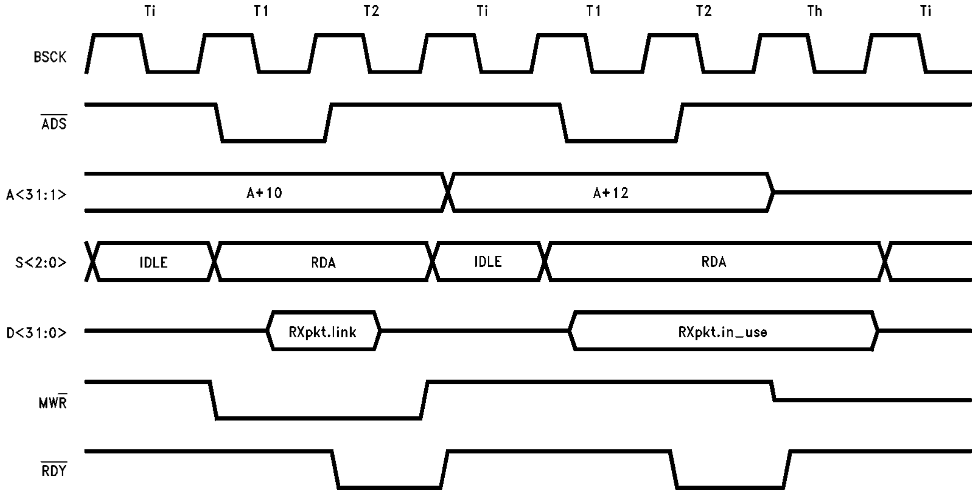
1.2.2 Detail of Access to the RDA

Figure 1-3a and *1-3b* shows the SONIC accessing the RDA for both BMODE = 0 and 1. Note that this block transfer contains both read and write accesses. Also note that the status pins S<2:0> briefly change from RDA (1,0,0) to Idle (1,1,1) between the read and write operations. This occurs between the RXpkt.seq__no and RXpkt.link accesses, and between the RXpkt.link and RXpkt.in__use accesses and is accompanied by a Ti bus clock state.



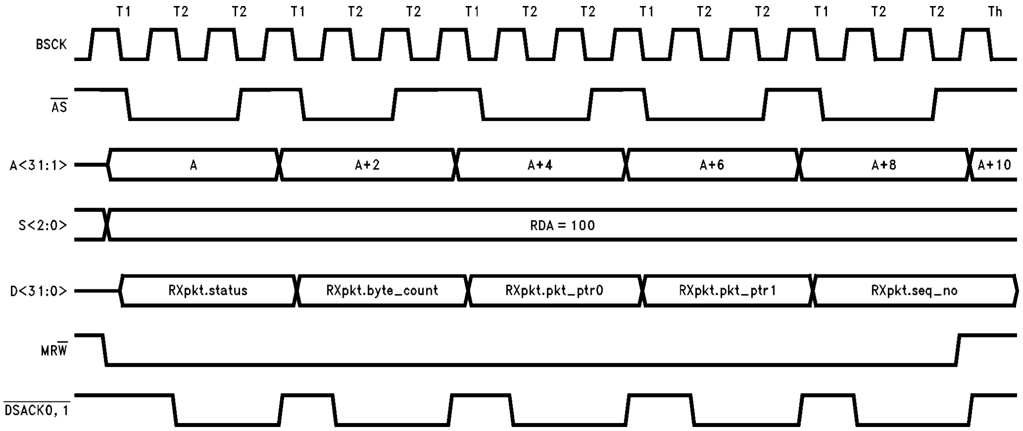
TL/F/11139–4

FIGURE 1-3a. RDA Access for Storing Descriptor Information (BMODE = 0, Synchronous Mode)



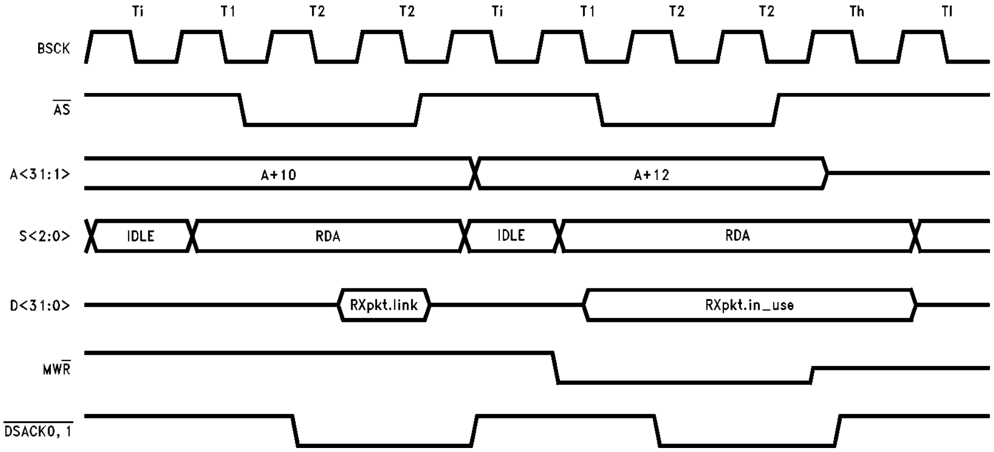
TL/F/11139–5

FIGURE 1-3a. RDA Access for Storing Descriptor Information (BMODE = 0) (Continued)



TL/F/11139–6

FIGURE 1-3b. RDA Access for Storing Descriptor Information (BMODE = 1, Asynchronous Mode)



TL/F/11139–7

FIGURE 1-3b. RDA Access for Storing Descriptor Information (BMODE = 1) (Continued)

1.3 Packet Transmission

This section gives a step-by-step analysis of a complete transmission using the initial conditions below.

Initial conditions:

(a) The Data Configuration register has been configured for:

- 32-bit data wide mode (DB5 = DW = 1)
- 16-byte Transmit FIFO threshold (BD1, 0 = TF1,0 = 0,1)
- Exact Block Transfer mode (DB4 = BMS = 1)

(b) The packet consists of 2 fragments. The first one is 48 bytes long and the last one is 20 bytes long.

(c) Both fragments are double-word aligned.

The transmit operation is described as follows: (the numbers in this section correspond to the numbers in *Figure 1-4*)

(1) Before the SONIC transmits, it fetches a descriptor from the Transmit Descriptor Area (TDA) to load its transmit registers. In 6 consecutive memory read operations, the SONIC reads the TXpkt.config, TXpkt.pkt_size, TXpkt.frag_count, TXpkt.frag_ptr0, TKpkt.frag_ptr1, and TXpkt.frag_size fields. Note that the TXpkt.status field is skipped during the first TDA access. Note also that if a collision occurs, forcing the SONIC to retransmit, the SONIC will once again fetch the descriptor from the beginning (i.e., starting at TXpkt.config).

(2) After fetching the descriptor, the SONIC begins loading the FIFO to its transmit threshold. The SONIC performs 4 consecutive memory operations in the Transmit Buffer Area (TBA) per bus tenure. Note that the fragment may begin on any byte boundary; if this is the case, the SONIC reads the corresponding double word which contains the beginning of the packet.

(3) The SONIC immediately requests for the bus again (in 3 bus clocks) because the number of words in the FIFO is equal to or less than the Transmit FIFO threshold. When the threshold has been exceeded, the SONIC commences transmission (TXE goes high). Subsequent requests for the TBA will not occur until the serializer has removed enough bytes from the FIFO to lower it below its threshold.

(4) Because of condition (b), the SONIC goes back to the Transmit Descriptor Area to obtain the pointer and length count of the next fragment. In three consecutive read operations, the SONIC will read the next TXpkt.frag_ptr0, TXpkt.frag_ptr1, and TXpkt.frag_size fields.

(5) At the end of transmission, the SONIC will write the status of the TXpkt.status field, then read the TXpkt.link field to locate the next descriptor.

1.3.1 Detail of Access to the TDA

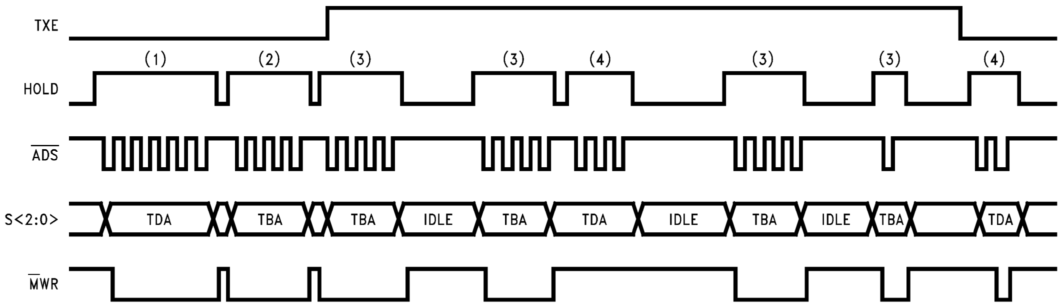
Figure 1-5a and *1-5b* shows the SONIC accessing the TDA at the end of transmission. The SONIC writes the status information at the beginning of the descriptor and reads the link field at the end of descriptor. (Note *n* = the number of fragments.) Since this access involves both a write and a read, there is a transition from TDA to Idle to TDA on the status lines in between writing the status and reading the link field. This transition is accompanied by a Ti bus clock state.

1.3.2 Detail of Access to the TBA

Figure 1-6a and *1-6b* shows the SONIC accessing the TBA when the transmit FIFO has been programmed for (1) 32-bit mode (2) exact block transfer mode, and (3) a 4 double word threshold.

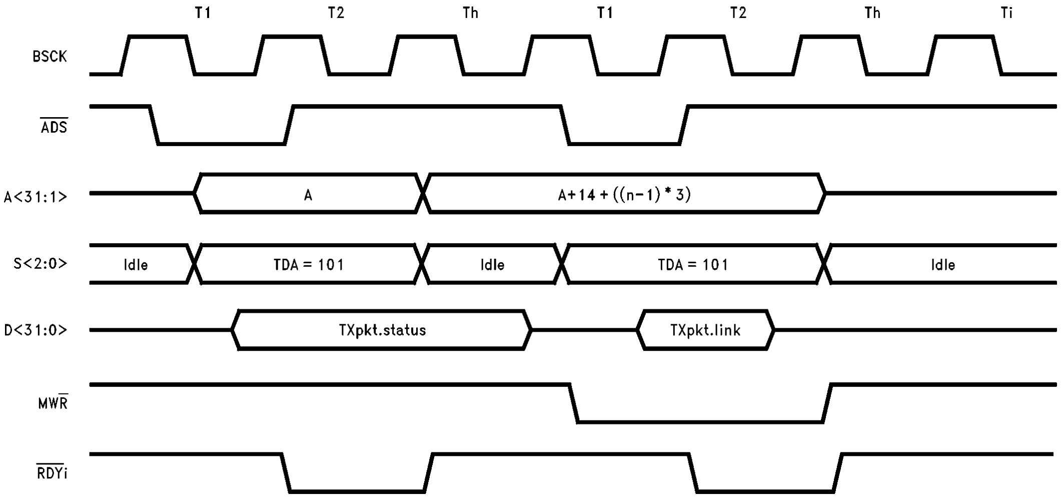
1.4 Loading the CAM (Content Addressable Memory)

After the CAM descriptor Area has been initialized and the Load CAM command issued to the SONIC, the SONIC will read the CAM Descriptor Area (CDA) and load its CAM. The SONIC, in 4 memory read cycles, accesses memory and loads one CAM entry per bus tenure. During the last block transfer, the SONIC reads one additional word to load its CAM Enable register. In the example illustrated in *Figure 1-7,* the SONIC has been programmed to load 4 CAM locations.



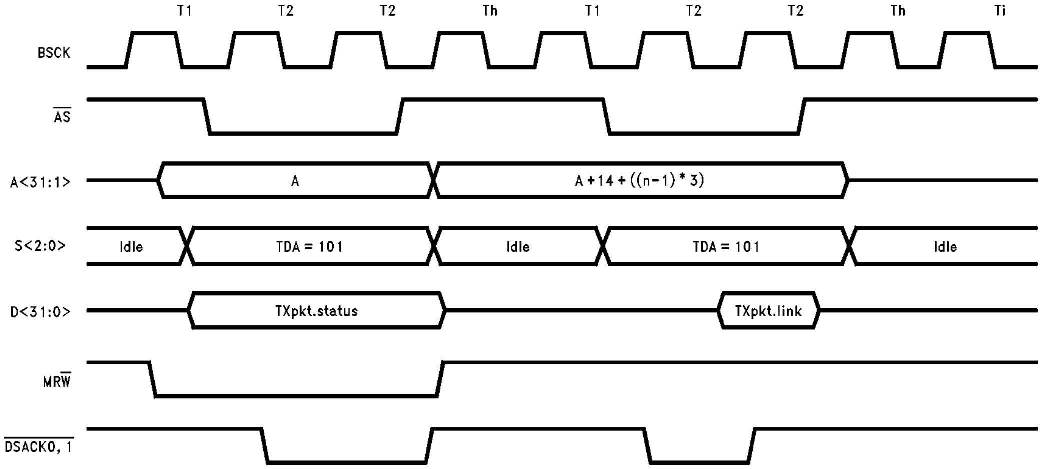
TL/F/11139–8

FIGURE 1-4. Complete Transmission of a 68-Byte Packet



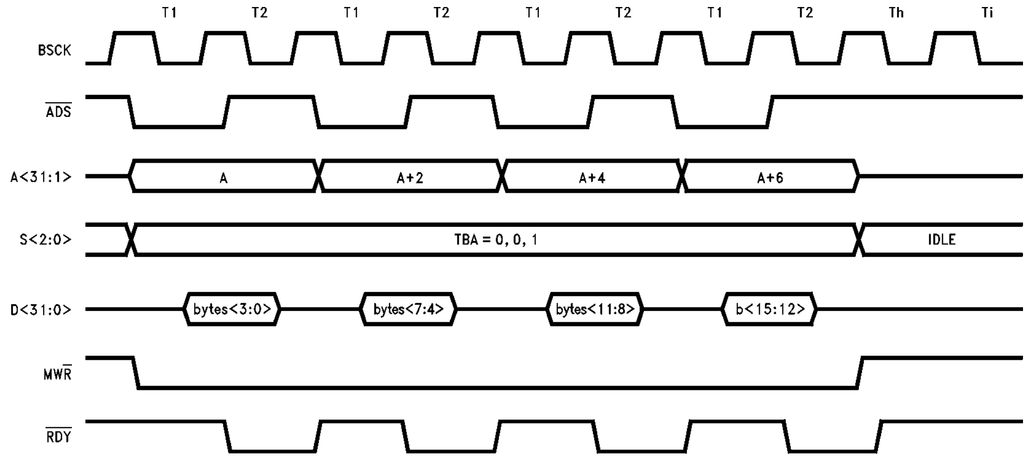
TL/F/11139–9

FIGURE 1-5a. Last TDA Access (BMODE = 0, Synchronous Mode)



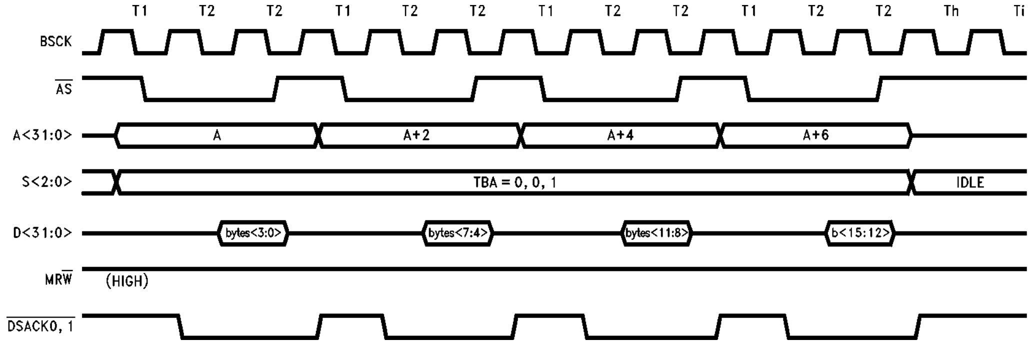
TL/F/11139–10

FIGURE 1-5b. Last TDA Access (BMODE = 1, Asynchronous Mode)



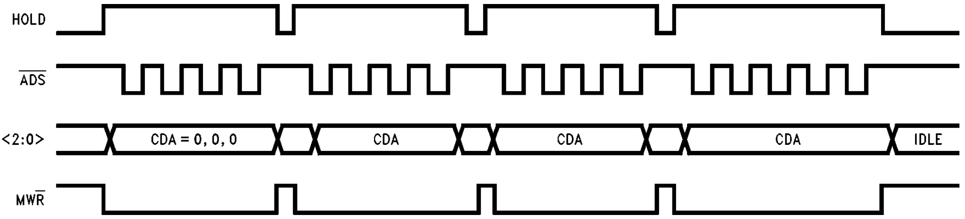
TL/F/11139–11

FIGURE 1-6a. Typical TBA Access (BMODE = 0, Synchronous Mode)



TL/F/11139–12

FIGURE 1-6b. Typical TBA Access (BMODE = 1, Asynchronous Mode)



TL/F/11139–13

FIGURE 1-7. Updating 4 CAM Entries with the Load CAM Command

2.0 SLAVE OPERATIONS

The slave operations of the SONIC can be classified into two cases, (1) register access while the SONIC is idle, and (2) register accesses while the SONIC is not idle. The first case always occurs in single bus systems where the CPU and SONIC reside on the same bus. Only one bus master is allowed on the bus in such a system. The second case may occur in dual bus systems where the CPU and SONIC lie on different busses. In this case, the CPU may access the SONIC while it is currently using the bus (such as during transmission or reception). The SONIC does not respond immediately in this case, but finishes off its current bus master operation before responding to the register access. The following two sections give a step-by-step description of the slave operations.

2.1 Register Access During Idle

(Refer to *Figures 2-1a* and *2-1b*)

(1) The CPU presents the register address, address strobe, chip select, and slave write/read strobe to the SONIC.

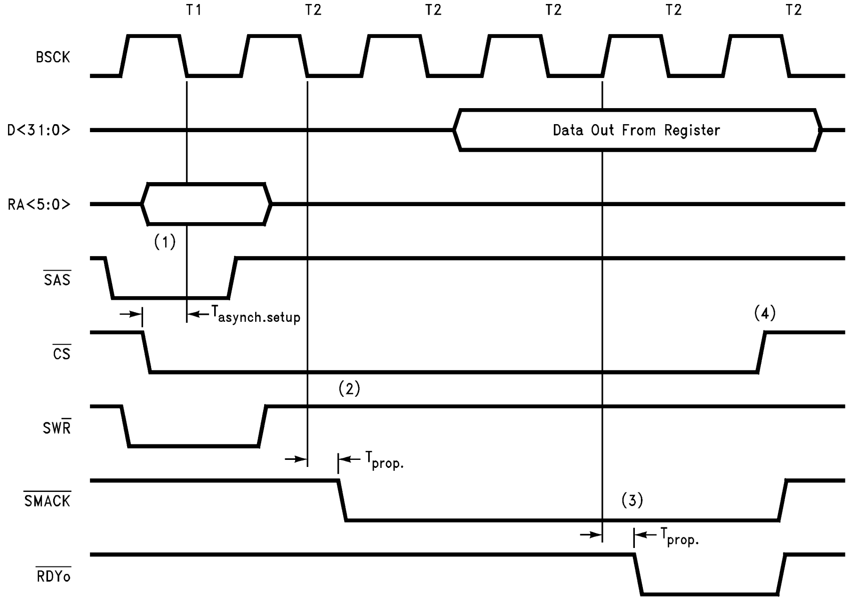
Note: For BMODE = 0, $\overline{\text{SAS}}$ must be asserted low before or at the same time that $\overline{\text{CS}}$ is asserted. The rising edge of $\overline{\text{SAS}}$ latches the register address, RA<5:0>, and slave direction strobe, SW$\overline{\text{R}}$.

(2) The SONIC synchronizes chip select to the falling edge of bus clock and responds with slave and memory acknowledge ($\overline{\text{SMACK}}$) at the next falling edge of bus clock.

Note: BMODE = 0, if $\overline{\text{SAS}}$ remains asserted (low) beyond the falling edge of $\overline{\text{CS}}$, $\overline{\text{SMACK}}$ will not be asserted until $\overline{\text{SAS}}$ is deasserted high.

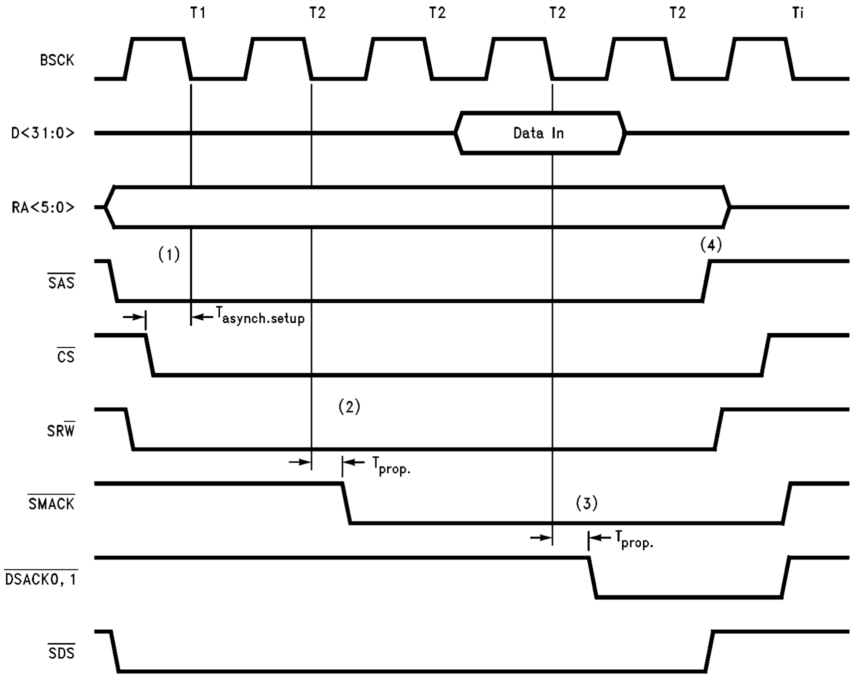
(3) For BMODE = 1, $\overline{\text{DSACK0,1}}$ is generated 2 bus clocks after $\overline{\text{SMACK}}$ is asserted, and for BMODE = 0, $\overline{\text{RDYo}}$ is generated 2.5 bus clocks after $\overline{\text{SMACK}}$. These outputs indicate to the CPU that the slave access is over. For read cycles, register data is valid at the falling edge of the ready signal ($\overline{\text{RDYo}}$ or $\overline{\text{DSACK0,1}}$); for write cycles, the SONIC has latched the register data at the falling edge of the ready signal.

(4) The CPU completes the slave cycle by deasserting $\overline{\text{CS}}$ or reasserting $\overline{\text{SAS}}$ low for BMODE = 0, or deasserting $\overline{\text{CS}}$ or $\overline{\text{SAS}}$ for BMODE = 1. The earliest of these signals will terminate the slave cycle.



TL/F/11139–14

FIGURE 2-1a. Register Read While SONIC is Idle (BMODE = 0)



TL/F/11139–15

FIGURE 2-1b. Register Read While SONIC is Idle (BMODE = 1)

2.2 Register Access While SONIC is a Bus Master

Figures 2-2a and *2-2b* show register accesses on the SONIC for both BMODE = 0 and BMODE = 1 respectively. These register accesses occur while the SONIC is a bus master (HOLD is high or $\overline{\text{BGACK}}$ is low). Notice how the bus states for the SONIC controlled DMA access (shown as states Ts1, Ts2 etc.) and the CPU controlled register access (shown as states Tc1, Tc2, etc.) overlap. They both start at the same time, but the register access does not complete until the SONIC DMA access completes and the SONIC gets off the bus. The following description refers to the numbers in *Figures 2-2a* and *2-2b*.

(1) In order to initiate a slave transfer, the SONIC must sample $\overline{\text{CS}}$ low. Also, it must sample $\overline{\text{SAS}}$ high for BMODE = 0 or $\overline{\text{SAS}}$ low for BMODE = 1. Even when the SONIC is doing master mode accesses on the bus, it monitors $\overline{\text{CS}}$ and $\overline{\text{SAS}}$. If the above conditions are met, the SONIC finishes the current master mode bus cycle and then TRI-STATES off the bus.

(2) The SONIC asserts slave and memory acknowledge ($\overline{\text{SMACK}}$) off the falling edge of the clock on the first Tc2 after the Tsh state.

(3) For BMODE = 1, $\overline{\text{DSACK0,1}}$ is generated 2 bus clocks after $\overline{\text{SMACK}}$ is asserted, and for BMODE = 0, $\overline{\text{RDYo}}$ is generated 2.5 bus clocks after $\overline{\text{SMACK}}$. These outputs indicate to the CPU that the slave access is over. For read cycles, register data is valid at the falling edge of the ready signal ($\overline{\text{RDYo}}$ or $\overline{\text{DSACK0,1}}$); for write cycles, the SONIC has latched the register data at the falling edge of the ready signal.

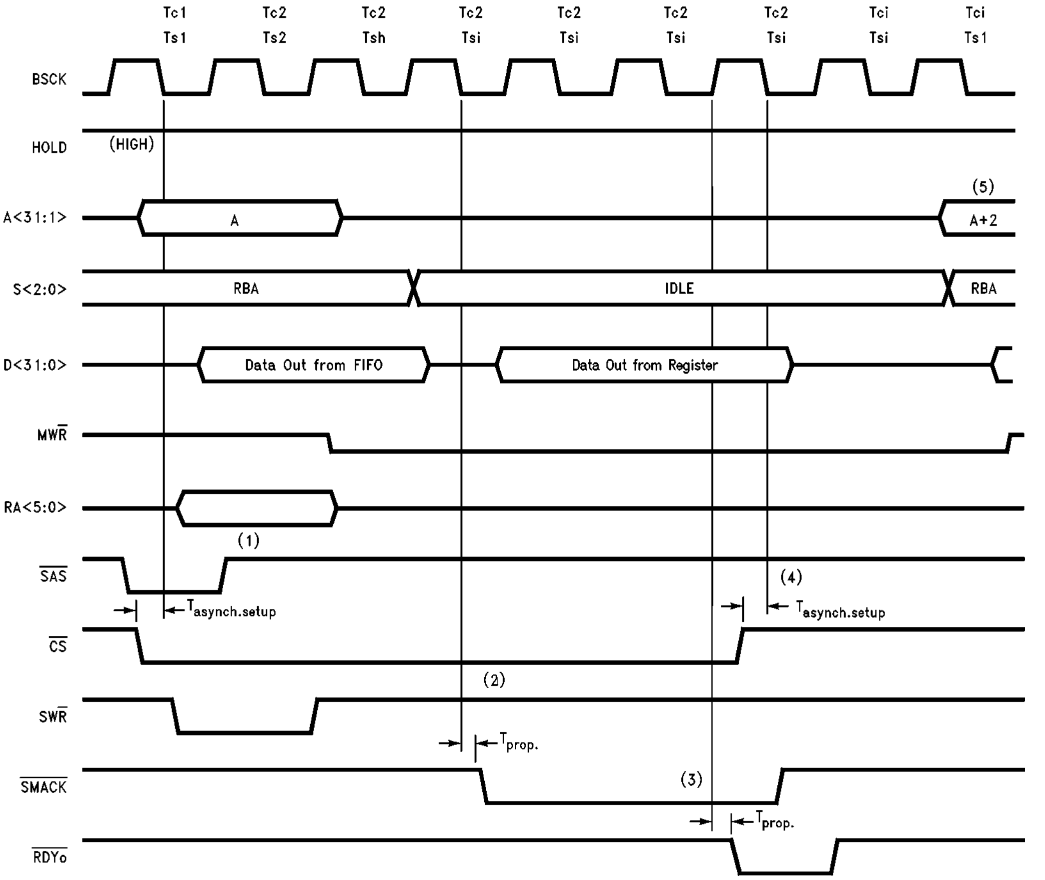
(4) The CPU terminates the slave cycle by deasserting chip select. Alternatively, the slave cycle can be terminated by driving $\overline{\text{SAS}}$ low for BMODE = 0, or $\overline{\text{SAS}}$ high for BMODE = 1.

(5) After the CPU has terminated the slave cycle, the SONIC continues its bus master operations from where it left off.

2.3 Asserting $\overline{\text{MREQ}}$ to Access Shared Memory

Asserting $\overline{\text{MREQ}}$ to the SONIC has nearly the same effect as asserting $\overline{\text{CS}}$. $\overline{\text{SMACK}}$ is generated identically as before, but the ready signal ($\overline{\text{RDY0}}$ or $\overline{\text{DSACK0,1}}$) is not asserted. The ready signal must be asserted by the memory control logic. Also, when using $\overline{\text{MREQ}}$, it is not necessary to assert $\overline{\text{SAS}}$.

Note: Both $\overline{\text{MREQ}}$ and $\overline{\text{CS}}$ must not be asserted simultaneously. This will cause spurious accesses to the SONIC's registers. Note also that the SONIC requires a recovery time of 2 bus clocks between the deassertion edge of one signal to the assertion edge of the other.



TL/F/11139–16

FIGURE 2-2a. Register Read While SONIC is Currently Using the Bus (BMODE = 0)

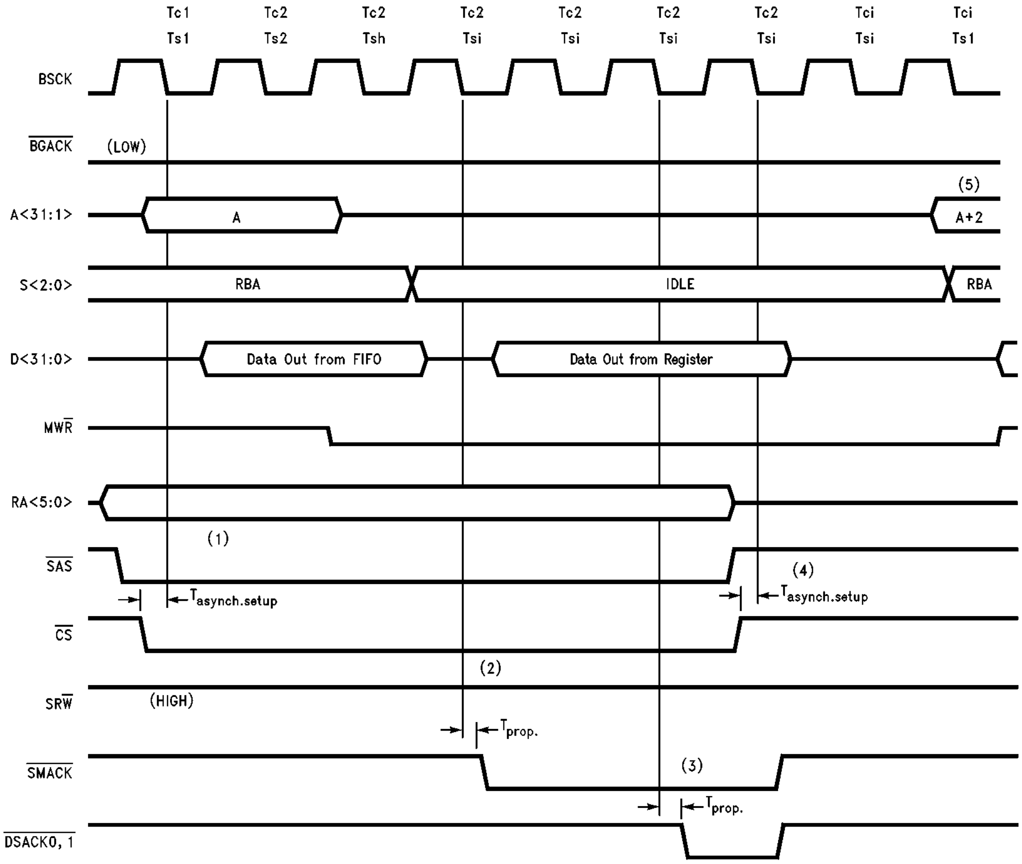


FIGURE 2-2b. Register Write While SONIC is Currently Using the Bus (BMODE = 1)

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
1111 West Bardin Road
Arlington, TX 76017
Tel: 1(800) 272-9959
Fax: 1(800) 737-7018

National Semiconductor Europe
Fax: (+49) 0-180-530 85 86
Email: cnjwge@tevm2.nsc.com
Deutsch Tel: (+49) 0-180-530 85 85
English Tel: (+49) 0-180-532 78 32
Français Tel: (+49) 0-180-532 93 58
Italiano Tel: (+49) 0-180-534 16 80

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 Semiconductor Japan Ltd.
Tel: 81-043-299-2309
Fax: 81-043-299-2408

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.