Integrated Device Technology, Inc.

INTERFACING THE R3051™ TO THE SONIC™

APPLICATION NOTE AN-95

by Danh Le Ngoc (Integrated Device Technology, Inc.) and Paul Cheng & Bill Harmon (National Semiconductor)

OVERVIEW

The IDT R3051™ family is a series of high-performance 32-bit microprocessors featuring a high-level integration and high-performance. The R3051 family integrates the MIPS R3000A™ RISC CPU, along with 8KB of instruction cache and 2KB of data cache. The R3051 family uses a simple time-multiplexed 32-bit address and data bus to provide a low cost system interface (and to minimize the cost of ASIC devices designed to interface with the processor). In order to minimize the impact of a time-multiplexed bus, the R3051 family incorporates a 4-deep read buffer and 4-deep write buffer into the interface, allowing relatively slow memory systems to be mated to a high-speed processor. The R3051 family is able to offer 35 MIPS of integer performance at 40MHz without requiring external SRAM or caches.

The R3051 family is designed to bring the high-performance inherent in the MIPS RISC architecture into low cost simplified embedded applications such as laser printers, X-Window terminals and network bridges and routers. Figure 1 illustrates the simplified block diagram of the R3051-based X-Window terminal.

The focus of this application note to describe the interface between the R3051 and National Semiconductor's System Oriented Network Interface Controller(SONIC).

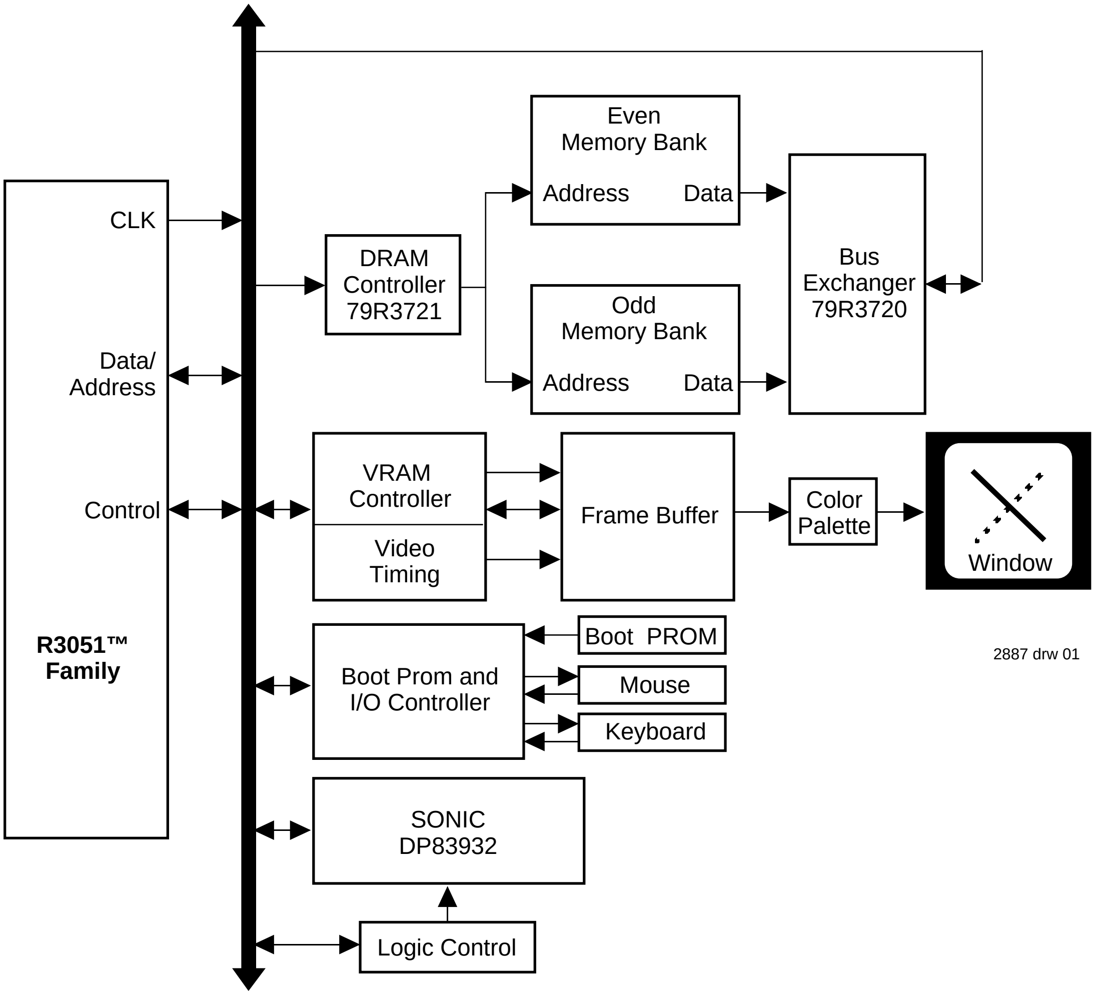


Figure 1. X-Window Terminal

SONIC is a Trademark of National Semiconductor
RISController is a Trademark of Integrated Device Technology

The SONIC™ is National Semiconductor's System Oriented Network Interface Controller (DP83932). This Ethernet controller is intended to provide a high performance 32 or 16-bit Ethernet connection for systems that require efficient, high throughput, low power network connectivity. The SONIC can be employed in an R3051-based system, in order to tightly couple the system's CPU and main memory to the network. Figure 2 depicts this interface.

The SONIC is ideally suited to embedded processing applications such as X-Terminals, due to its unique feature set. The SONIC completely supports all the required specifications set forth in the IEEE 802.3 standard, including the Media Access Control (MAC) requirements contained in the IEEE 802.3 layer management specification. Additionally, SONIC's high performance DMA channels allow it to use a very small percentage of the bus bandwidth, while its efficient linked list buffer management scheme limits the number of descriptor and data fetches required. It is also important to note that the SONIC utilizes internal content addressable memory (CAM) to provide a 100% perfect address filter for both multicast and physical address packets. This alleviates the need to waste bus bandwidth, memory space, and CPU time on unwanted packets. Finally, the SONIC contains an integrated Manchester encoder/decoder, which is required in all Ethernet applications. This provides a savings in board space, as well as improved reliability.

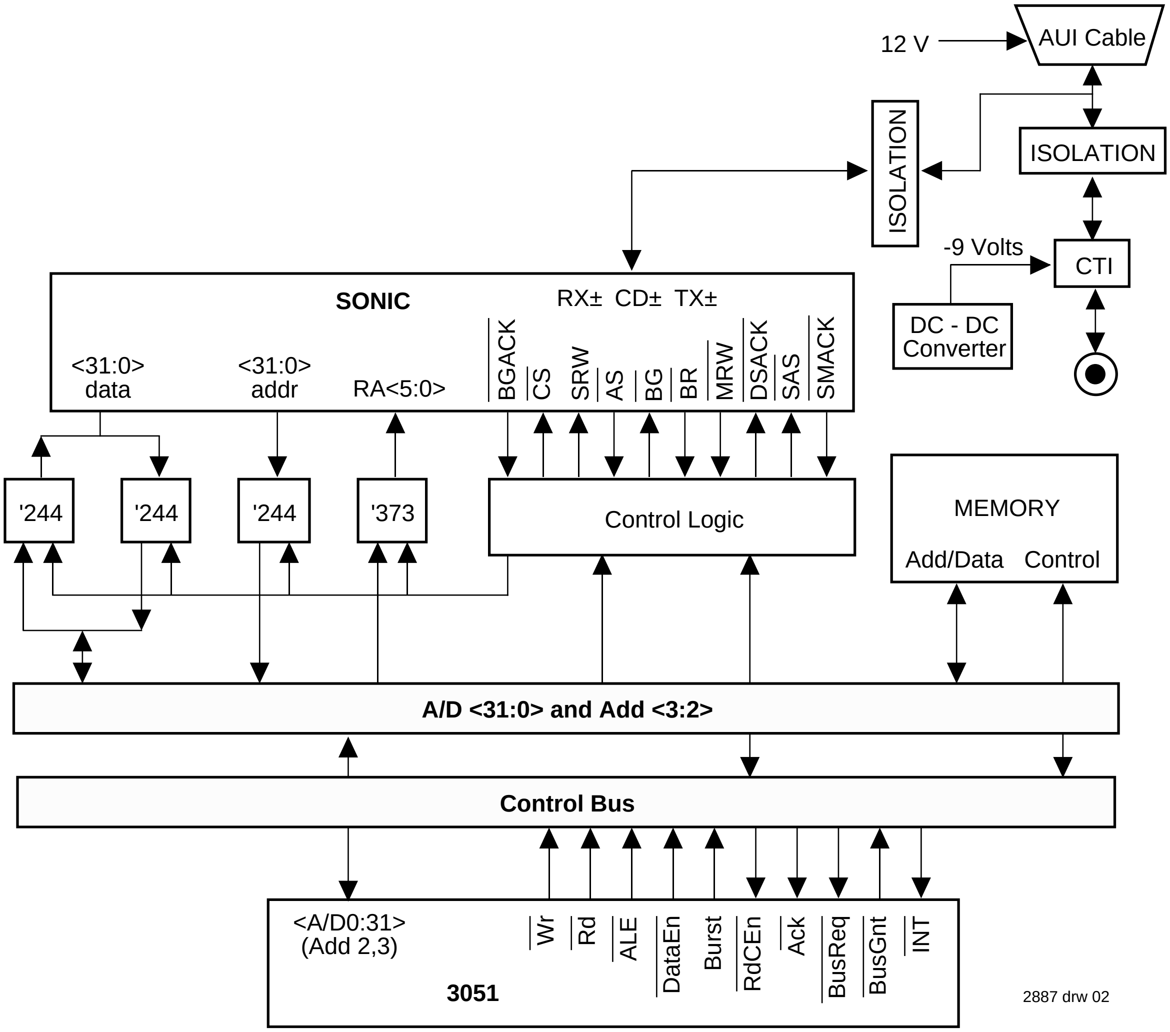


Figure 2. SONIC Interface to the R3051

FUNCTIONAL OVERVIEW

System Interface

The R3051 has a multiplexed 32-bit address and data bus. Since the SONIC's address and data buses are demultiplexed, it is necessary to employ a set of external latches to connect the SONIC to the processor's address and data buses. In many applications, these latches may also be used to demultiplex the R3051 bus to other parts of the system memory and I/O.

In order to allow the R3051 to have access to the SONIC's internal registers, as well as allow the SONIC to gain control of the system bus and perform DMA operations, the SONIC is interfaced to the system bus as both a slave and a master. As a slave, the SONIC appears as a block of 256 bytes, consisting of sixty-four 32 bit words. The SONIC can be mapped into any location of memory and will typically provide for a 7 cycle register access. In R3051 applications, the SONIC will typically be mapped into the processor kseg1, which is an unmapped, uncached address space typically used for processor I/O resources.

As a master, the SONIC will arbitrate with the R3051 for ownership of the bus and proceed to operate as a 32-bit DMA engine between the network and the system memory. While operating on the bus, the SONIC is capable of performing 32-bit/3 cycle DMA operations. It is important to note that the ability to place the SONIC on the same bus as the R3051 and the system memory is critical: this eliminates the need for the Ethernet controller to have a local buffer, which the CPU must spend time and bandwidth to transfer to main memory. The ability of the SONIC to place data directly in main memory and communicate with the CPU through linked list descriptors, as well as register accesses, makes the SONIC/R3051 interface CPU and bandwidth efficient.

Network Interface

With respect to the physical layer design, both AUI drop cable Ethernet and thin wire Ethernet are supported. The block diagram in Figure 2 contains a 15 pin AUI drop cable connector for standard drop cable Ethernet implementations, as well as a thin wire Ethernet connection via the National Semiconductor coaxial transceiver interface (CTI, DP8392). Either of these network connections can be chosen through the use of a single jumper between the 5 volt supply and the 5 volt to –9 volt DC-to-DC converter. In either case, the AUI signals (RX±, TX±, and CD±) are sent back to the SONIC. These signals are interfaced to the ENDEC portion of the SONIC, which provides for communication between the AUI interface and the non-return to zero (NRZ) signals (RXD,TXD, and COL) of the Media Access Control (MAC) module of the SONIC. It should be noted that the integrated ENDEC module of the SONIC alleviates the need for an external Ethernet Manchester encoder/decoder, such as National's CMOS Serial Network Interface (CMOS SNI, DP83910).

ARCHITECTURE AND DESIGN

Bus Interface

The SONIC's bus interface can be externally configured to operate in one of two modes. If the SONIC's BMODE pin is tied to ground, the SONIC will operate on the bus exactly like an 80386 microprocessor. If the SONIC's BMODE pin is tied to 5 volts, the SONIC will operate on the bus exactly like a 68030 microprocessor. In this design, the most appropriate mode of operation was achieved by connecting BMODE to 5 volts.

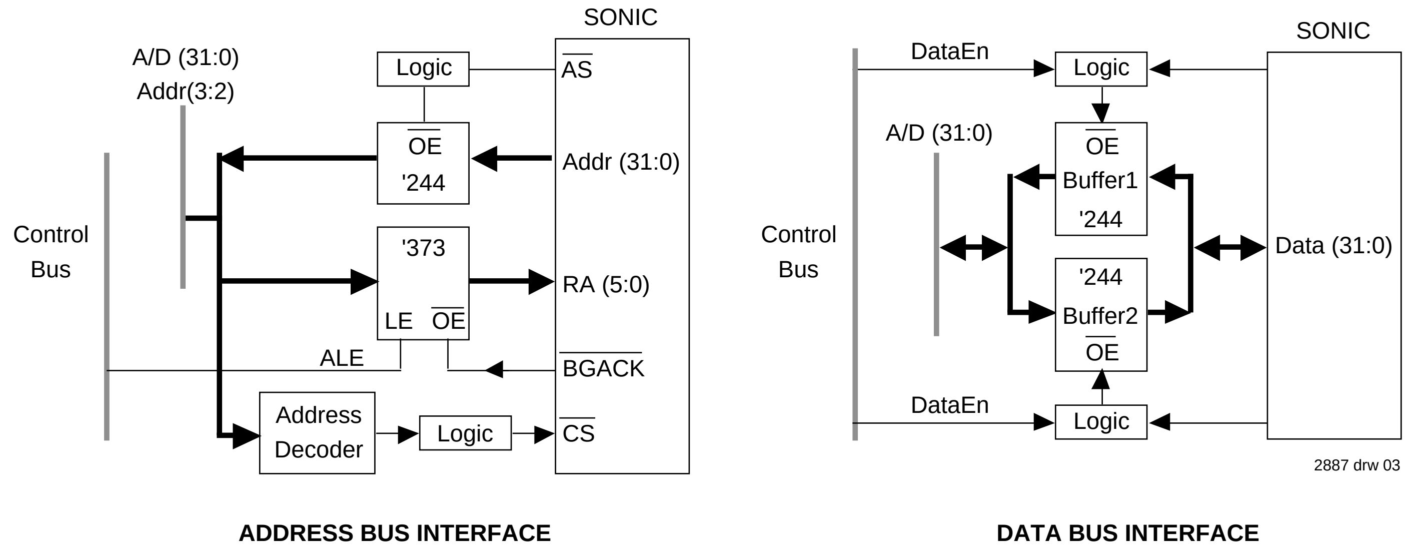
The bus interface, as depicted in Figure 3, consists of 2 parts. There is an address bus interface and a data bus interface. Since the R3051's address and data buses are multiplexed, it is necessary to utilize a set of '244 buffers and '373 latches to multiplex the SONIC busses onto the CPU bus. The '244 buffers are required to tri-state the SONIC's address lines from the system bus during the data portion of master transfers, while the '373 is required to latch the register addresses being sent to the SONIC during slave operations. The output enable signal of the '244 is asserted when the SONIC is the master of the bus and both the SONIC's address strobe ($\overline{\text{AS}}$) is asserted and the master logic's address latch enable (ALE) signal is asserted. The '373 should latch the address when the R3051 is the bus master and it asserts its ALE signal.

The data bus interface requires the use of 2 sets of '244 buffers. The first set of buffers (Buffer 1) prevent the SONIC from placing data onto the system's multiplexed address and data bus prematurely. In the slave mode of operation, the output buffer is enabled once the address output drivers are tri-stated. This is signaled by the assertion of the DataEn signal. In the case of a master operation, the buffers are enabled once the address buffers external to the SONIC are tri-stated, which takes place upon the deassertion of the ALE signal.

The second set of buffers is enabled when the SONIC's registers are being written by the R3051 and data is being presented on the multiplexed system address/data bus, or when the SONIC is reading system memory and the memory is placing data on the multiplexed address / data bus. The assertion of the DataEn signal by the system signals that data is now able to be placed on the bus. The actual logic representation for the bus interface can be found in the bus interface logic segment of the Control Logic section of this application note.

Slave Operation

The timing diagram for a slave access of the SONIC is shown in Figure 4. The falling edge of the R3051's ALE signal latches the output of an address decoder and the address lines being passed to the register address lines of the SONIC. If the address decode selects the SONIC, a signal called "AdrDec" will be asserted. The logic for generating this signal



ADDRESS BUS INTERFACE

DATA BUS INTERFACE

Figure 3. Address and Data Bus Interface

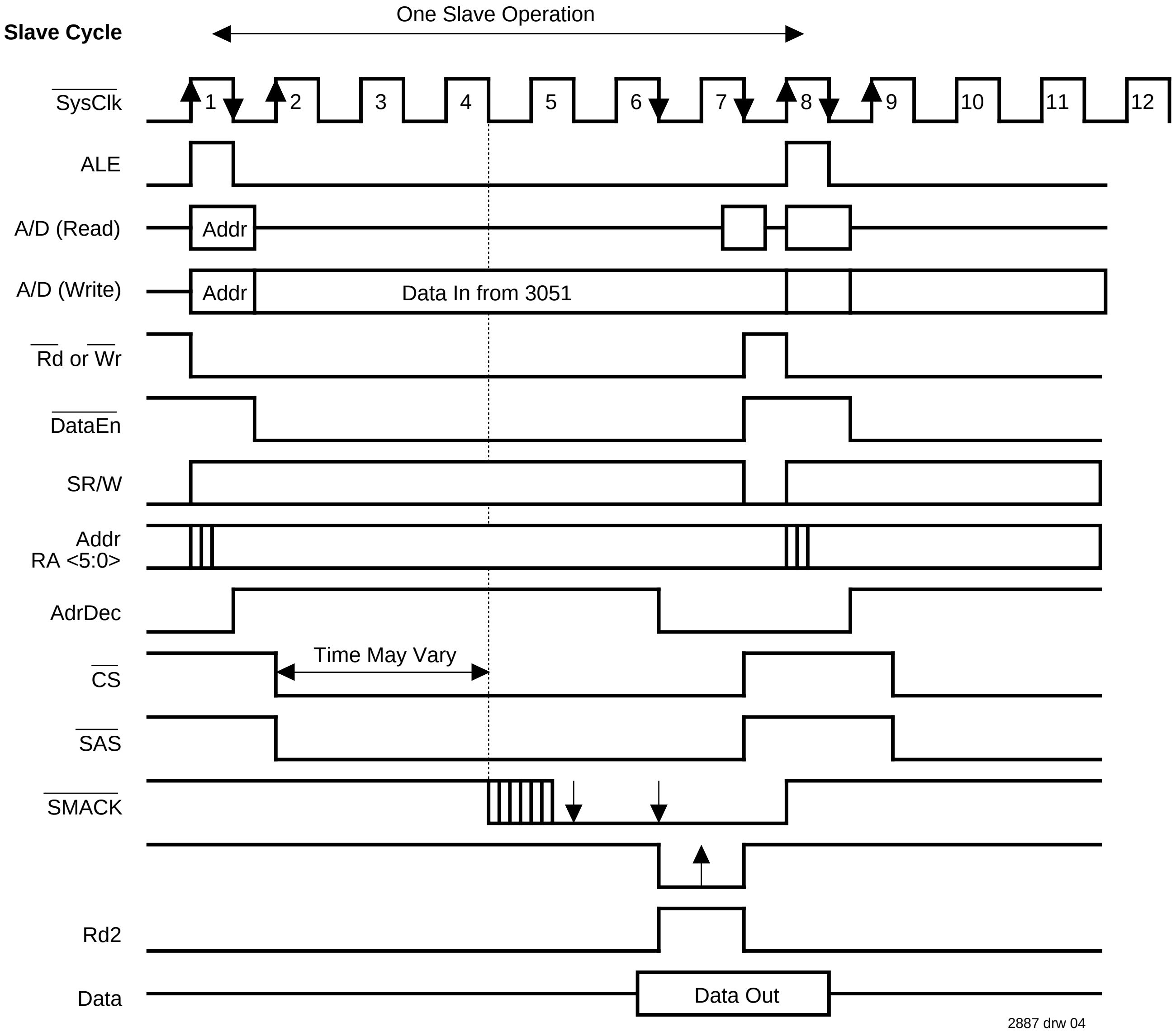


Figure 4. Slave Access Timing Diagram

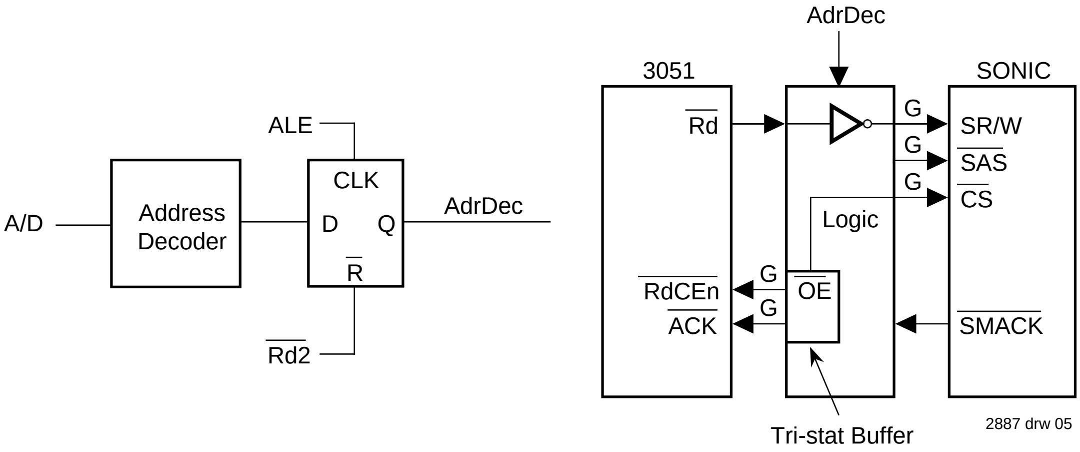


Figure 5. Slave Interface Block Diagram

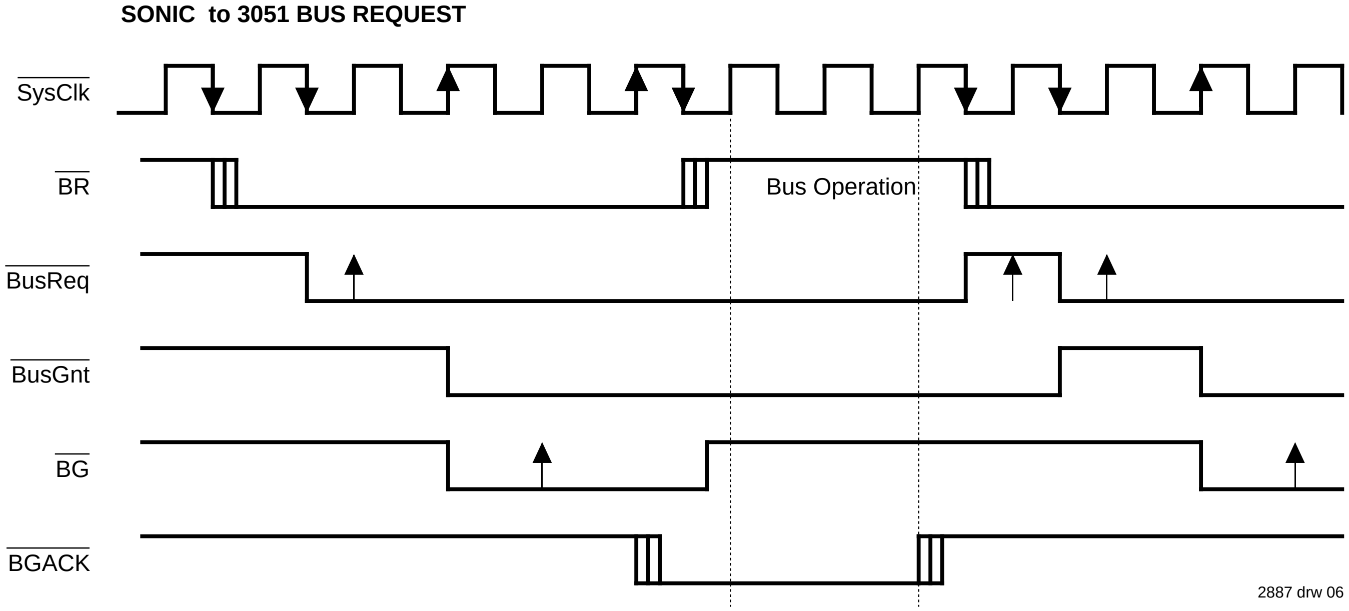


Figure 6. Bus Request Timing Diagram

is shown in Figure 5. The value of this signal is passed to the chip select (CS) and slave address strobe (SAS) signals of the SONIC on the rising edge of the bus clock. The acknowledge signals back to the R3051 (ACK for a write and RdCEn for a read) are asserted 2 clocks after the SONIC generates its slave acknowledge signal (SMACK). These signals remain asserted to the R3051 for a clock cycle, after which they are removed. The ACK and RdCEn signals inform the R3051 that the data has been latched or is valid, respectively. The deassertion of these signals results in the deassertion of CS and SAS to the SONIC. The logic for implementing this part of the design can be found in the slave logic segment of the Control Logic section.

Master Operation

The first step in designing the master interface is implementing the bus request logic. The timing diagram for this is shown in Figure 6. The bus request (BR) signal of the SONIC is passed to the R3051's bus request (BusReq) on the falling edge of the bus clock. The SONIC then waits for the bus grant (BusGnt) from the R3051, which is passed directly to the SONIC's bus grant (BG) signal. The assertion of BG causes the SONIC to assert bus grant acknowledge (BGACK) and begin its master DMA operations. It is important to note that the assertion of BGACK causes the SONIC to deassert BR, which would cause the bus request logic to deassert BG to the SONIC. Thus, the BusReq signal to the R3051 should be the logical "OR" of the SONIC BR and BGAck outputs. A block diagram of the bus request logic appears in Figure 7, while the actual illustration of the logic is found in the bus request logic segment of the Control Logic section.

Once the SONIC has gained control of the bus, it will begin to perform master DMA operations, as illustrated in the Figure 9 timing diagram. Ideally, if the memory is fast enough, the SONIC will be able to perform 3 cycle DMA. At 25 MHz, less than 3.75% of the bus' bandwidth will be consumed by the network interface.

There are two very important points to note. First, the R3051's ACK signal is basically equivalent to the SONIC's DSACK signals, but the SONIC's DSACK signals require that the memory system provide a total of 8 ns hold time from the rising edge of the clock, while the R3051 requires only 4 ns. Second, the ALE signal generated from the SONIC's control signals will be deasserted 3 ns later than the R3051's would be. However, this should not be a significant factor, since the address set-up and hold time provided to the memory system's latches is consistent with the R3051's specification.

When interfacing to the multiplexed bus, it is necessary for the master logic to generate an ALE signal for the system bus. The ALE signal is asserted on the rising edge of the second cycle in the SONIC's memory access. It is necessary to assert the ALE in this cycle, in order to guarantee that the latch will be provided with an adequate amount of set-up time for the address. The ALE signal is then removed on the falling edge of the same clock cycle. The deassertion of ALE triggers the assertion of $\overline{\text{DataEn}}$ on a read operation, in order to inform the memory that the bus' address drivers are tri-stated and data can now be driven. The $\overline{\text{DataEn}}$ signal is actually arrived at by delaying the the ALE signal through a buffer or PAL, since the ALE signal is also responsible for disabling the output buffers of the address drivers.

The final piece of interface logic is used to make the SONIC's read and write (MR/W) strobe compatible with the R3051's read ($\overline{\text{Rd}}$) and write ($\overline{\text{Wr}}$) signals. The SONIC's read/ write signal is passed to the appropriate read or write strobe

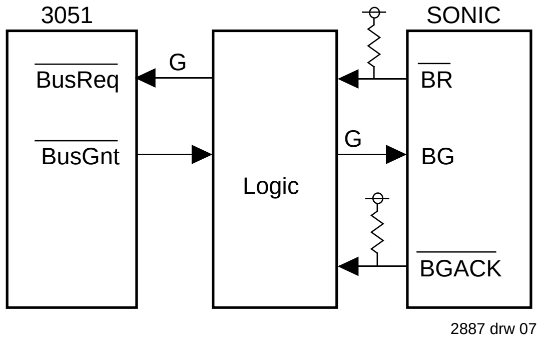


Figure 7. Bus Request Interface Block Diagram

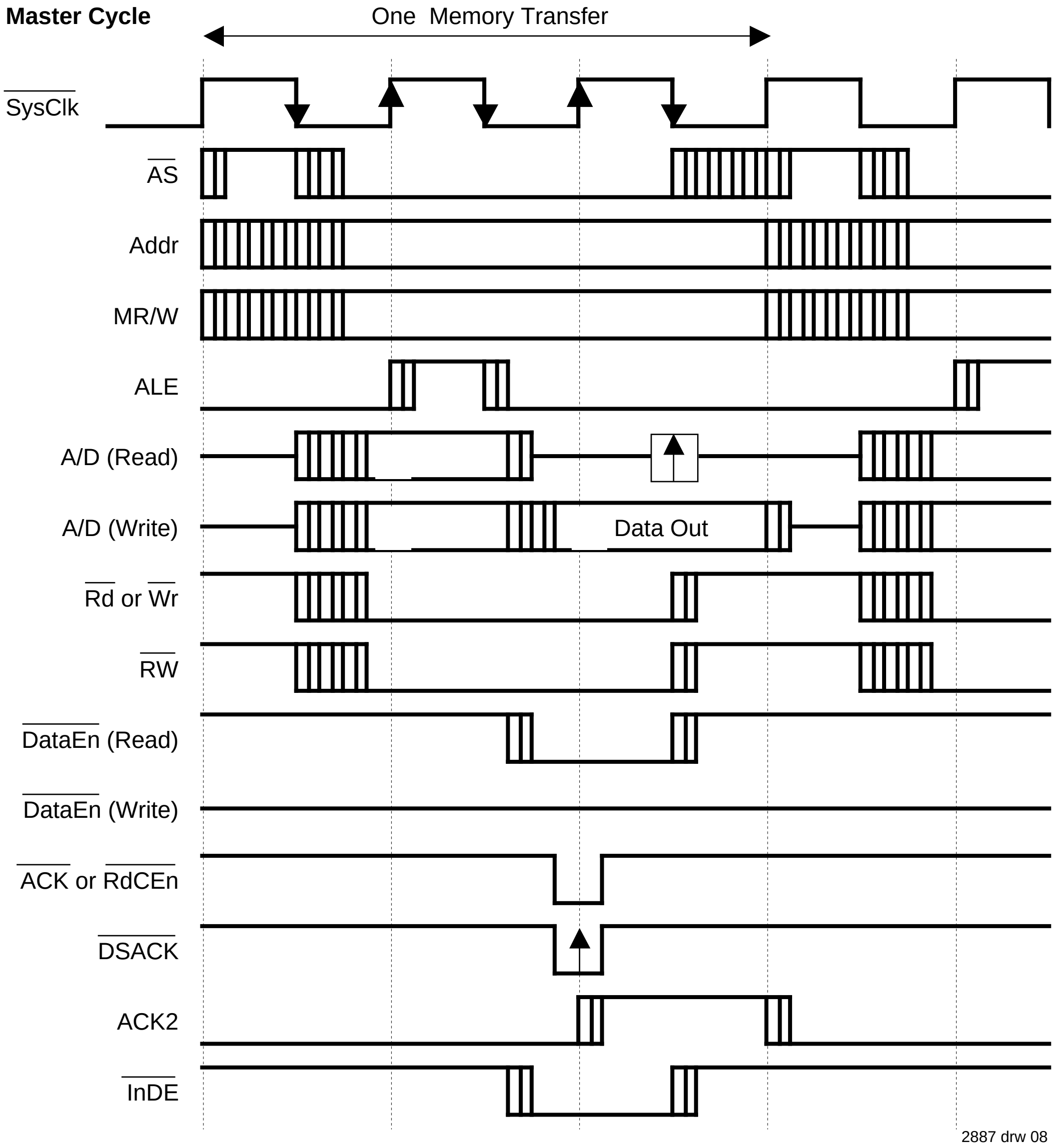


Figure 8. Master Access Timing Diagram

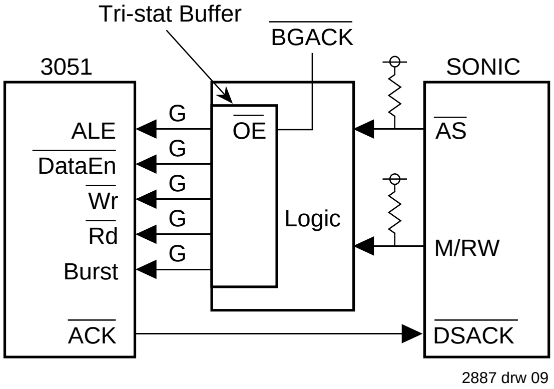


Figure 9. Master Interface Block Diagram

of the system bus, on the falling edge of AS. The Rd or Wr signal is then deasserted on the falling edge of the last clock cycle. The block diagram for the master interface is found in Figure 9, while the logical implementation is shown in the master interface logic segment of the Control Logic section.

Physical Layer

Figure 10 contains a block diagram of the physical layer interface, while a schematic of the physical layer design is located on the last page of this application note. This design can be used in either a thin wire or standard drop cable Ethernet environment. When the design is used in a thin wire Ethernet application, the 5 volt supply must be connected to the DC-to-DC converter, so that the necessary –9 volt output can be supplied to National Semiconductor's Coaxial Transceiver Interface (CTI, DP8392). The CTI provides an interface between the 10 MHz Manchester encoded coax cable and the 10 MHz Manchester encoded differential signals of the SONIC's ENDEC. In the case of a standard drop cable Ethernet application, the 5 volt supply is left unconnected, so that the CTI will not receive power. This allows the signals of the SONIC's ENDEC to pass directly to the AUI cable, via the 15 pin AUI connector. In examining the schematic of the physical layer design, it can be seen that there is a pulse transformer at the AUI side of the CTI. This is placed here to isolate the CTI from the SONIC's ENDEC signals, when the AUI drop cable connection is being employed. This transformer also provides the IEEE 802.3 specified isolation between the coax and the differential AUI signals, when thin wire Ethernet is being used. It is also necessary to provide a termination for the 78Ω AUI cable's differential receive and collision pair (RX± and CD ±). This is the reason for the 39Ω –1%resistors and .01µF capacitors that are shown in Figure 10.

Additionally, there are 2 more significant considerations. First, each one of the transmit pairs (TX+ and TX-) requires a 270Ω non-precision pull down resistor to complete the internal source follower amplifiers that drive these signals. Second, there is an isolation transformer placed between the differential signals of the SONIC's ENDEC and the AUI cable. This isolation is necessary to guarantee that the SONIC meets the IEEE 802.3 fail safe specification of a 16V DC level appearing on the AUI cable's differential signals. This external isolation is necessary, because in the powered down state the CMOS process, in which the SONIC is manufactured, may not be able to withstand this voltage.

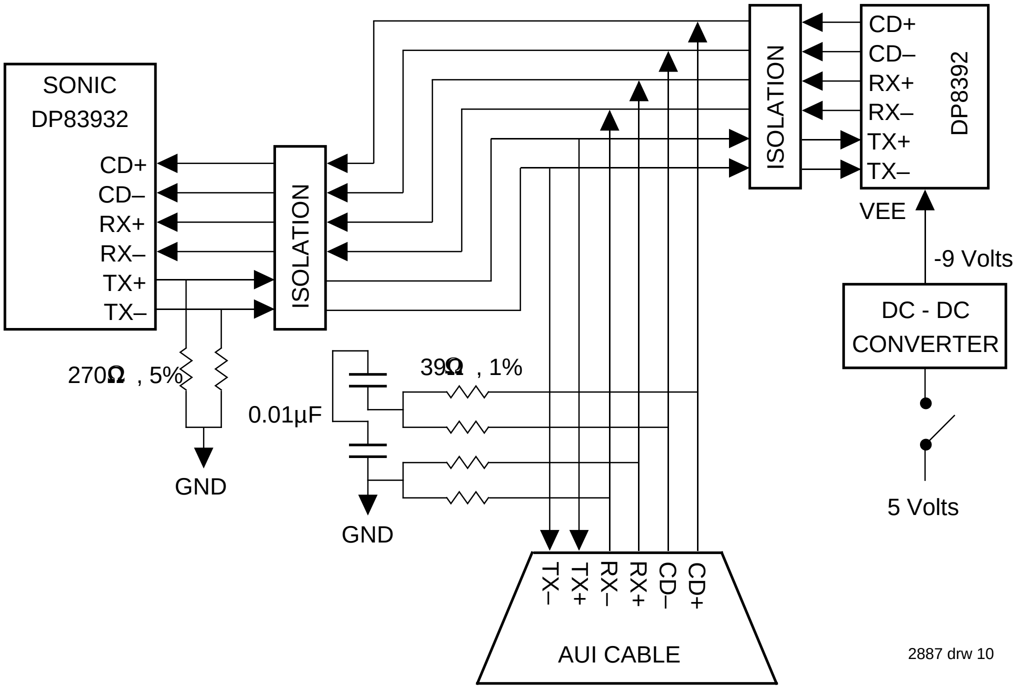


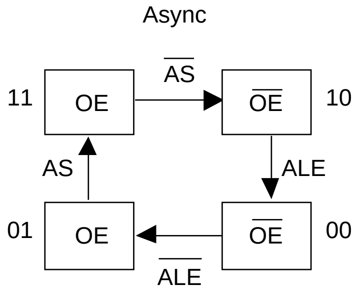
Figure 10. Physical Layer Interface Block Diagram

Control Logic

This application note was developed with the intention of displaying the necessary requirements for interfacing the SONIC to the R3051 system bus. Therefore, the actual implementation of the control logic will be graphically depicted in state machine form, as opposed to being partitioned into actual PAL devices. This leaves the freedom for the designer to incorporate this logic into his / her system in PALs, ASICs, FPGAs, etc.

BUS INTERFACE LOGIC

Address



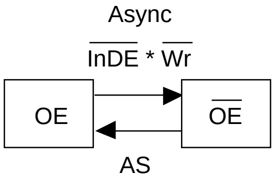
$$Q1^{\wedge} = Q1 * Q2 + Q1 * \overline{ALE} + Q2 * AS$$

$$Q2^{\wedge} = \overline{Q1} * Q2 + \overline{Q1} * \overline{ALE} + Q2 * AS$$

$$OE = Q2^{\wedge}$$

Data

$$OE = \underbrace{(\overline{DataEn} + CS)}_{\text{1st case}} + \underbrace{(BGACK + DataEn)}_{\text{2nd case}}$$



$$OE1 = DataEn + CS$$

$$OE2 = Q^{\wedge} = AS * InDE + AS * \overline{Q} + Wr * Q + InDE * Q$$

$$OE = \underbrace{DataEn + CS}_{\text{case 1}} + \underbrace{AS * InDE + AS * OE2 + Wr * \overline{OE2} + InDE * OE2}_{\text{case 2}}$$

2887 drw 11

Note:

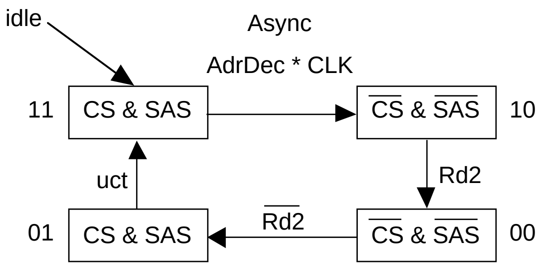
1. Q1^ refers to the first state machine bit and Q2^ refers to the second state machine bit (10: Q1^=1 & Q2^ = 0)

SLAVE INTERFACE LOGIC

$Q1^{\wedge} = Q2 + Q1 * \overline{Q2} * \overline{Rd2}$

$Q2^{\wedge} = Q1 * Q2 * \overline{AdrDec} + \overline{Q1} * \overline{Q2} * \overline{Rd2} + \overline{Q1} * Q2$

$CS = SAS = Q2^{\wedge}$

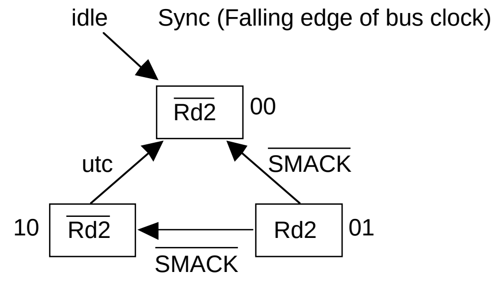


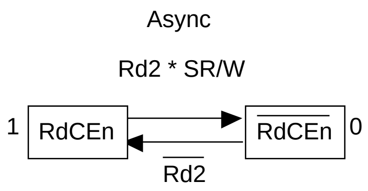
idle
Async
AdrDec * CLK
11
CS & SAS
$\overline{CS}$ & $\overline{SAS}$
10
uct
Rd2
$\overline{Rd2}$
01
CS & SAS
$\overline{CS}$ & $\overline{SAS}$
00

$Q1^{\wedge} = \overline{SMACK} * Q2$

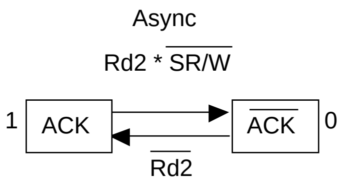
$Q2^{\wedge} = \overline{Q1} * \overline{Q2} * \overline{SMACK} + Q2 * SMACK$

$Rd2 = Q2^{\wedge}$





$RdCEn = Q^{\wedge} = \overline{Rd2} + Q * \overline{SR/W}$



$ACK = Q^{\wedge} = \overline{Rd2} + Q * SR/W$

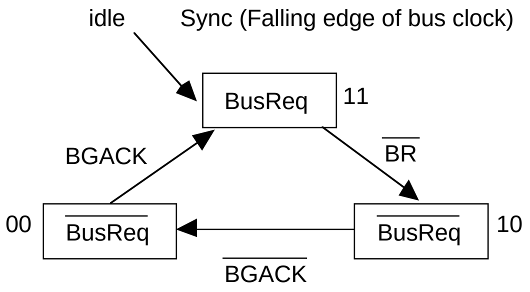
2887 drw 12

BUS REQUEST INTERFACE LOGIC

$Q1^{\wedge} = BGACK + Q1 * Q2$

$Q2^{\wedge} = BR * Q2 + BGACK * \overline{Q1}$

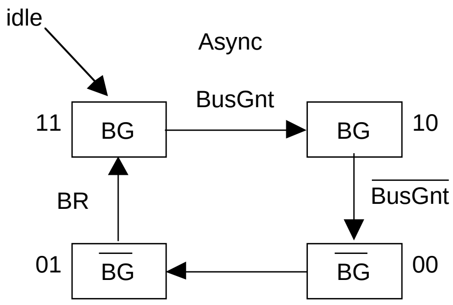
$BusReq = Q1^{\wedge} * Q2^{\wedge}$



$Q1^{\wedge} = Q1 * Q2 + Q2 * BR$

$Q2^{\wedge} = \overline{Q1} + Q2 * \overline{BusGnt}$

$BG = Q1^{\wedge}$



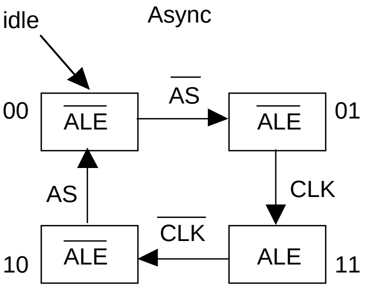
2887 drw 13

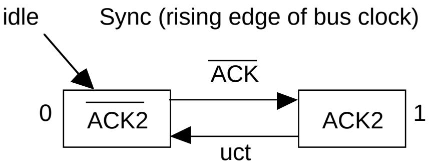
MASTER INTERFACE LOGIC

$Q1^{\wedge} = Q1 * \overline{AS} + Q2 * CLK + Q1 * Q2$

$Q2^{\wedge} = \overline{Q1} * \overline{AS} + \overline{Q1} * Q2 + Q2 * CLK$

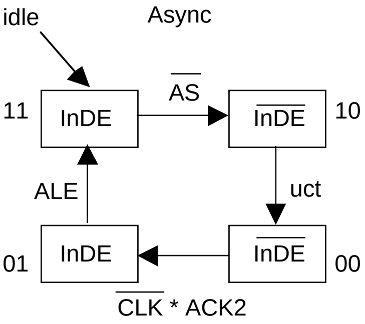
$ALE = Q1^{\wedge} * Q2^{\wedge}$

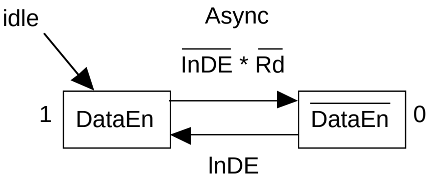




$Q^{\wedge} = \overline{Q} * \overline{ACK}$

$ACK2 = Q^{\wedge}$





$Q1^{\wedge} = Q1 * Q2 + \overline{Q1} * Q2 * ALE$

$Q2^{\wedge} = \overline{Q1} * Q2 + \overline{Q1} * \overline{Q2} * \overline{CLK} * ACK2$

$+ Q1 * Q2 * ALE$

$InDE = Q2^{\wedge}$

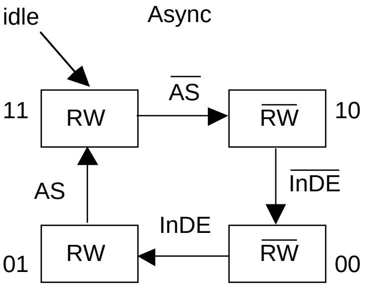
$Q^{\wedge} = InDE + Rd * Q$

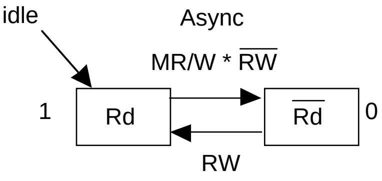
$DataEn = Q^{\wedge}$

$Q1^{\wedge} = Q1 * Q2 + Q1 * InDE + AS * Q2$

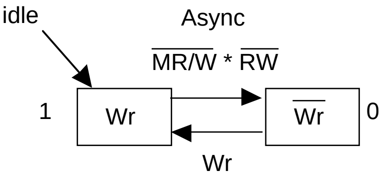
$Q2\ ^{\wedge} = InDE * \overline{Q1} + \overline{Q1} * Q2 + Q2 * AS$

$RW = Q2^{\wedge}$





$Rd = Q^{\wedge} = RW + Q * \overline{MR/W}$



$Wr = Q^{\wedge} = RW + Q * MR/W$

2887 drw 14