DATAPOINT

ARCNET

Designer's Handbook

61610

Preface

ARCNET® is the local area networking scheme used by the Datapoint Attached Resource Computer® (ARC®) system, a multiple processor system designed for resource sharing and modular expansion. The network provides intercommunication among up to 255 nodes at a rate of 2.5 Megabits per second.

Although conceived as an integral part of the overall ARC system, ARCNET has proved to be a viable local area network for equipment of any manufacturer. This document is provided as an aid to those considering the use of the ARCNET protocol in their own equipment.

The custom integrated circuits described in this document are manufactured by various vendors under license with DATAPOINT and are available to the public from those vendors. For a current list of vendors, contact DATAPOINT Product Marketing, Local Area Networking Group, 512/699-7151.

CONTENTS

CHAPTER 1.
ARCNET OVERVIEW

Contents

PROTOCOL

ARCNET is a token passing system. Control, including detection and re-creation of lost tokens, is completely dispersed among the nodes of the network. The system is dynamically configurable, allowing nodes to enter and leave the network independently with negligible impact on performance. Deterministic parameters are provided by means of the even rotation of the token among the nodes coupled with a maximum time allotment for any operation at any node.

HARDWARE

Processors interface to ARCNET via special purpose communications adapters called RIMs (Resource Interface Modules). The RIM chip is an NMOS LSI implementation of the major portions of the RIM.

The ARCNET protocol imposes very few restrictions on the transmission medium. Basically, the receiver at any node must be able to hear the transmitter at any other node, any two nodes must be connected by a single path, and the path between any two nodes must have a propagation delay of 31 microseconds or less.

The STANDARD ARCNET INTERFACE is a baseband system using dipulse signaling on RG-62 coax. The usual disadvantage of a baseband system is the difficulty of tapping the line for distribution. This problem is circumvented through the use of 'hubs.' Each node connects, through a length of coax, to a port on a hub; the coax is properly terminated at each end and no taps of any kind are used. In a sense, the hub may be thought of as an amplifier and a number of ideal taps all mounted in the same box! These virtual taps are ideal in the sense that they have no insertion loss and no tap loss, and yet provide total suppression of reflections, even from unterminated lines. Each of the ports of a hub may be connected to

an ARCNET node, to another hub, to an unterminated length of coax, or to nothing at all.

SOFTWARE

Interaction between the processor and the RIM is on a message by message, as opposed to a byte by byte, basis. The RIM contains four 256 byte message buffers and the arbitration logic required to share them between the processor, the RIM transmitter, and the RIM receiver. To send a message, the processor writes the message into a RIM buffer and issues a transmit command; the RIM sets a status bit (and can cause an interrupt) when the message has been sent. To receive a message, the processor assigns a RIM buffer to the RIM receiver; the RIM sets a status bit (and can cause an interrupt) when a message has been received and is available in the buffer. The user is totally unaware of, and has no control over, any of the details of the token passing protocol.

CHAPTER 2.
ARCNET PROTOCOL

Contents

LINE PROTOCOL

The line idles in a spacing (logic 0) condition. A transmission starts with an ALERT BURST consisting of six unit intervals of mark (logic 1). Eight bit characters are then sent with each character preceded by two unit intervals of mark and one unit interval of space. Five types of transmission are sent:

INVITATITONS TO TRANSMIT: an ALERT BURST followed by three characters; one EOT (End Of Transmission) and two (repeated) DID (Destination IDentification) characters. Used to pass control of the line from one RIM to another.

FREE BUFFER ENQUIRIES: an ALERT BURST followed by three characters; one ENQ (ENQuiry) and two (repeated) DID (Destination IDentification) characters. Used to ask a RIM if it is able to accept a packet.

PACKETS: an ALERT BURST followed by from 8 to 260 characters; one SOH (Start Of Header), one SID (Source IDentification), two (repeated) DIDs (Destination IDentification), one CONTINUATION POINTER, from 1 to 253 data characters, and two CRC (Cyclic Redundancy Check) characters. Used to move data between RIMs.

ACKNOWLEDGMENTS: an ALERT BURST followed by one character; an ACK (ACKnowledgment). Used to acknowledge PACKETS and as an affirmative response to FREE BUFFER ENQUIRIES.

NEGATIVE ACKNOWLEDGMENTS: an ALERT BURST followed by one character; a NAK (Negative AcKnowledgment). Used as a negative response to FREE BUFFER ENQUIRIES.

The receiver validates all incoming transmissions by checking for:

- at least one mark and exactly one space preceding each character,
- an EOT, ENQ, SOH, ACK, or NAK following the ALERT BURST,
- proper CRC (packets only),
- proper number of characters (3, 8 to 260, or 1),
- at least nine spaces following the last character.

LINK CONTROL

Each RIM in a system has a unique ID (IDentification) from 1 to 255, which is selected by jumpers.

Note: Although 256 possible identifications exist, ID 0 may not be assigned to any RIM since destination 0 is used to indicate a BROADCAST to all RIMs.

System operation is based on an INVITATION TO TRANSMIT being passed around the system with each RIM passing the INVITATION TO TRANSMIT to the NID (Next ID), which is the RIM with the next higher ID in the system. When a RIM receives an INVITATION TO TRANSMIT containing its ID, that RIM assumes control of the line. The RIMs that are in the system are determined during SYSTEM RECONFIGURATION.

When a RIM is first turned on or when it has not received an INVITATION TO TRANSMIT for approximately 840 milliseconds (ms), it causes a SYSTEM RECONFIGURATION by sending a RECONFIGURE BURST consisting of eight marks and one space repeated 765 times. The purpose of this burst is to terminate all activity on the line. This RECONFIGURE BURST is longer than any other type of transmission and will therefore interfere with the next INVITATION TO TRANSMIT by preventing any RIM from seeing the INVITATION and assuming control of the line.

The RECONFIGURE BURST also provides line activity so that the RIM sending the INVITATION TO TRANSMIT releases control of the line. Thus the RIM that had control releases it and no other RIM picks it up. When any RIM sees idle line for 78 microseconds (μs), it knows the system is being reconfigured and initializes its NID to its own ID. The RIM then starts a time out equal to 146 μs times 255 minus its own ID [146 μs x (255 -ID)]. If this time-out expires with no line activity, the RIM begins sending

INVITATIONs TO TRANSMIT. (Note that this time-out will expire only in the RIM with the highest ID in the system.)

After sending an INVITATION TO TRANSMIT, the RIM waits for activity on the line (e.g., the RIM receiving the invitation sending a FREE BUFFER ENQUIRY, PACKET, or INVITATION TO TRANSMIT or any RIM sending a RECONFIGURATION BURST). If there is no activity for 74 μs, the RIM increments NID and tries again. If it hears any activity before the time-out expires, the RIM releases control of the line.

During SYSTEM RECONFIGURATION, INVITATIONs TO TRANSMIT will be sent to all 256 possible IDs. Each RIM, however, will have saved NID, the ID of the RIM that assumed control from it. From then until the next SYSTEM RECONFIGURATION (which will occur only when a new RIM is powered up or when a RIM is dropped from the system due to line errors causing it to miss an INVITATION TO TRANSMIT), control is passed directly from RIM to RIM with no wasted INVITATIONs TO TRANSMIT being sent to IDs not in the system.

The time required for a SYSTEM RECONFIGURATION varies from 24 to 61 ms, depending on the number of RIMs in the system and the propagation delays between them.

DATA EXCHANGE

When a RIM receives an INVITATION TO TRANSMIT, it checks to see if it has a packet to send, i.e. it checks to see if TA (the Transmitter Available status bit) is false. If not, it sends an INVITATION TO TRANSMIT to NID. Otherwise, the RIM tests byte 001 in the transmit buffer, which is the DID (Destination IDentifier). If this byte is 000, the packet is a BROADCAST and the RIM sends the packet; otherwise, it sends a FREE BUFFER ENQUIRY to the DID RIM and waits up to 74 μs for a response. If the response to the FREE BUFFER ENQUIRY is an ACK, the RIM sends the packet. If the response to the FREE BUFFER ENQUIRY is a NAK, the RIM sends an INVITATION TO TRANSMIT to NID and will send another FREE BUFFER ENQUIRY the next time it receives an INVITATION TO TRANSMIT. If the RIM times out waiting for a response to the FREE BUFFER ENQUIRY, it sets TA and sends an INVITATION TO TRANSMIT to NID.

After sending a packet, the RIM waits up to 74 μs for a response. If it receives an ACK, it sets TMA (the Transmitted Message Acknowledged status bit), and then sets TA. Finally, it sends an INVITATION TO TRANSMIT to NID. If the RIM times out waiting for an ACK (packets are never NAK'ed), it only sets TA and sends an INVITATION TO TRANSMIT to NID.

When a RIM receives a FREE BUFFER ENQUIRY, it tests RI (the Receiver Inhibited status bit). If RI is true, it sends a NAK; otherwise, it sends an ACK.

When a RIM receives an SOH (indicating the start of a PACKET), it writes the SID into the receive buffer and then checks the first DID. If the first DID is 000 (the PACKET is a BROADCAST) and reception of BROADCASTs has been enabled, or if the first DID is the RIM's own ID, the RIM writes the second DID and the rest of the PACKET into the receive buffer. If these conditions are not met, the RIM ignores the rest

of the PACKET. If, after being written into the receive buffer, the PACKET fails either the CRC or length validation phases, the RIM ignores it. If the PACKET passes these phases, the RIM tests byte 001 in the receive buffer, the DID. If this byte is 000, the packet is a BROADCAST and the RIM simply sets RI. If this byte is the RIM's own ID, the RIM sends an ACK before setting RI. If the DID is neither 000 nor the RIMs own ID, the RIM ignores the packet.

PERFORMANCE FACTORS

The most interesting aspect of performance in a local network is usually the amount of time a node may have to wait before being able to send a message. In a token passing scheme, this waiting time is bound by the time it takes the token to make the rounds of each and every node. There are several possible sequences of events that can occur when a node receives the token. Two of these make up the vast majority of cases; the others have to do with error conditions, messages sent to nonexistent nodes, etc. The two cases of interest are a simple token pass and a message followed by a token pass.

A simple token pass takes between 28 and 59 μs, depending on the propagation delay between the two nodes. (In even the largest real world systems, the propagation delay between nodes, averaged over a complete token trip, is such that the average time per token pass is very close to 28 μs.) A message followed by a token pass takes between 141 and 296 μs, depending on propagation delays, plus 4.4 μs per byte of data. Thus the time required for the token to make a complete trip around the network is approximately 28 μs per node plus 113 μs per message plus 4.4 μs per byte.

In a system, for example, consisting of 10 nodes and sending messages no longer than 100 bytes, a complete token trip will be at least 280 μs (when no messages are sent) and no longer than 5810 μs (when all 10 nodes send a 100 byte message). If only a single node is sending messages, it can send one every 833 μs, giving it a rate of 1200 messages, or 120000 bytes, per second. If all 10 nodes are sending messages, each one can send one every 5810 μs, a rate of 172 messages per second per node.

At the other end of the size spectrum, the original ARCNET system in the DATAPOINT Research and Development Department now has anywhere from 150

to over 200 nodes active at any given time. It supports two totally independent operating systems and a wide variety of uses including program loading, word processing, print spooling, program development, electronic mail, etc. The traffic load rarely falls below 400 messages per second, yet less than 2% of the nodes send a message on the average token trip. The time required for a token trip, therefore, stays very close to the no traffic value. Peaks of three times the no traffic value are extremely rare.

CHAPTER 3.
ARCNET HARDWARE

Contents

ANATOMY OF A RIM

A complete RIM consists of four basic sections:

- **CONTROLLER** - the RIM chip interfaced to 1k bytes of RAM buffer

- **CLOCK** - the clock oscillator and drivers

- **LINK INTERFACE** - the circuitry required to interface the CONTROLLER to the serial data link using media such as baseband coax, broadband coax, fiber optic cable, etc.

- **PROCESSOR INTERFACE** - the address decoders, bus drivers, etc. required to interface the CONTROLLER to the system bus of the host processor.

CONTROLLER

The CONTROLLER section consists of the RIM chip, an eight-bit node ID switch, a shift register, a 1K x 8 static RAM, and a gated latch. Control signals from the RIM chip periodically load the switch into the shift register and then shift (most significant bit first) the node ID into the RIM chip. The eight-line Address-Data bus and two-line Page bus provide the address and data path to the PROCESSOR INTERFACE. The Address-Data bus is multiplexed between address and data under control of the RIM chip.

BUFFER MEMORY AND CONTROL

The buffer memory is composed of 1K of static RAM. The memory is segmented into four pages, 256 bytes each. It is a dual port memory, accessed directly by the RIM chip, or indirectly by the processor under control of the RIM chip. The RIM chip's bus control logic provides the control strobes for both access modes.

During a transmit sequence, the RIM chip reads data from the transmit buffer, one of the four 256-byte pages of the buffer memory. The particular page used is selected by a two-bit field in the transmit command written into the RIM chip command register. During a receive sequence, the RIM chip writes data in the receive buffer, which is also one of the four 256-byte pages of the buffer memory. The particular page used is selected by a two-bit field in the receive command written into the RIM chip command register.

When the processor is accessing the RAM, the RAM address and the data are exchanged directly with the processor interface. The RIM chip only generates the control signals.

A suitable memory is the MOSTEK MK4801AN-4 RAM. The RAM's chip select pin ($\overline{CS}$) is grounded, and RIM chip address output A8 is not used. An external latch latches the lower eight bits of the address.

BUS ARBITRATION AND CONTROL LOGIC

Since the buffer is a two-port memory, arbitration logic is required to govern access to the memory. The Address-Data bus, AD7-AD0, provides a path for three separate types of data transfer:

- RIM chip access to buffer memory

- Processor access to buffer memory

- Processor access to RIM chip control registers.

The RIM chip requests occur when the transmitter or receiver needs to read or write the buffer. The processor request can result from either a buffer memory access request or from a control/status I/O command.

The two types of processor data transfers cannot occur simultaneously, so arbitration between these is unnecessary. RIM chip memory requests are asynchronous to both types of processor requests, however, so arbitration is required.

The bus arbitration logic allocates bus cycles for the three separate functions. In the event of simultaneous requests from the processor and the RIM chip, the processor has priority. At the end of a processor cycle, the RIM chip will be granted use of the bus, if the chip is waiting.

CLOCK

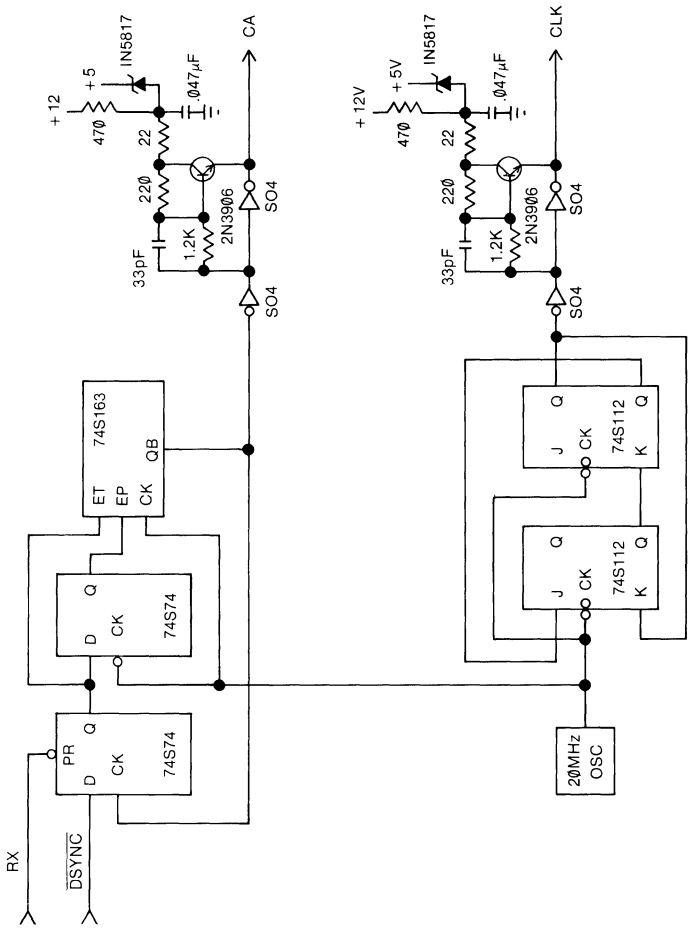
The CLOCK section provides two 5 MHz clock signals. One of these is free running and is applied to the CLK inputs of the RIM chip and the ID shift register. The other clock is synchronized to the received data under control of the $\overline{\text{DSYNC}}$ output of the RIM chip and applied to the CA input of RIM chip.

Synchronization is accomplished by sampling $\overline{\text{DSYNC}}$ on the rising edge of CA and stopping the clock when $\overline{\text{DSYNC}}$ is found to be low. This will occur during the ALERT BURST and the stop element of each received character. CA is left high while the clock is stopped.

The clock is restarted when the received signal from

the LINK INTERFACE (the RX input to the RIM chip) goes low as the start element of a character is received. When the clock is restarted, CA should go low 100 nanoseconds (ns), nominally, after the leading edge of the start element, so that the rising edge of CA occurs in the center of the start element.

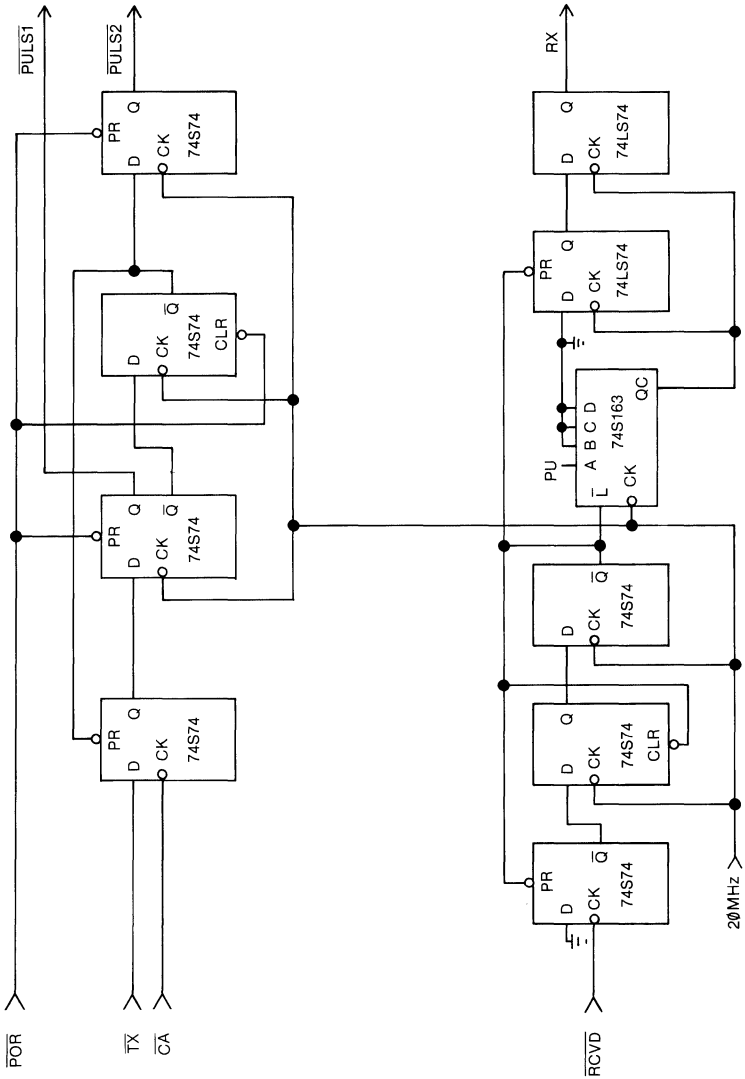
The clock section may be implemented either with TTL components or with the TRANSCEIVER chip, a companion to the RIM chip.



Typical Clock Generator Circuitry

LINK INTERFACE

The LINK INTERFACE section of the RIM interfaces the serial data link to the CONTROLLER section via the $\overline{TX}$ and RX pins of the RIM chip. The exact implementation of this section depends on the type of data link (twisted pair, coax, etc.) and signaling scheme (NRZ, Manchester, etc.) used. The version described here is the STANDARD ARCNET INTERFACE; a baseband system using dipulse signaling on RG-62 coax and allowing runs of up to 2000 feet. The link interface may be implemented either with TTL logic and discrete analog components or with the TRANSCEIVER chip and the ARCNET Interface Hybrid.



LINK INTERFACE

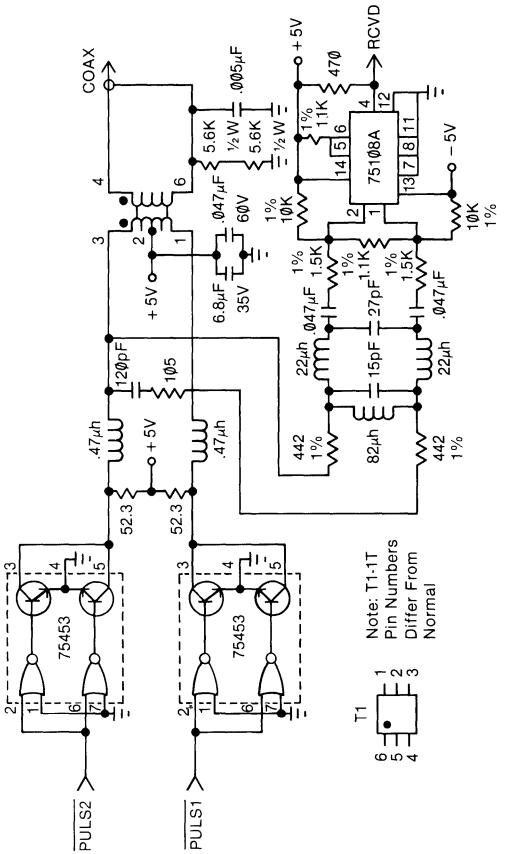
TRANSMITTER

The serial output of the RIM chip is $\overline{TX}$. This pin remains high for spaces and goes low for 200 nanoseconds for marks.

The STANDARD ARCNET INTERFACE samples $\overline{TX}$ on the trailing edge of CA and generates two 100 nanosecond wide pulses for each mark. These negative going pulses, $\overline{PULSE1}$ and $\overline{PULSE2}$, are used to drive the line interface.

ARCNET LINE INTERFACE (Standard ARCNET Implementation)

The two 100 nanosecond pulses from the transmitter generate a 200 nanosecond wide dipulse on the coax. Incoming pulses on the coax are passed through a matched filter and line receiver.



LINE INTERFACE

RECEIVER

The output of the line receiver is converted to NRZ. A low-to-high transition at the line receiver output sets a filp-flop. The flip-flop is sampled at 50 nanosecond intervals. When a one is detected, the receiver resets the flip-flop and generates a 400 nanosecond pulse on RX. (RX is the serial input of the receiver in the RIM chip. The data format at this pin is NRZ with spaces low and mark high.)

PROCESSOR INTERFACE

The processor sees the RIM as two separate sets of contiguous eight bit registers, or memory locations:

- the two location set of control/status in the RIM chip,
- the 1024 location set in the buffer memory.

The PROCESSOR INTERFACE section of the RIM provides the interface necessary to allow the host processor to access these two areas. Depending on the requirements of the particular processor and the whims of the hardware designer and/or system programmer, these areas may appear as I/O registers, locations in memory space, locations in I/O space, or some combination of these.

The interface is asynchronous, with timing controlled by AS and $\overline{\text{DWR}}$. Processor access cycle requests begin on the trailing edge of AS if either $\overline{\text{IOREQ}}$ or $\overline{\text{MREQ}}$ is asserted. The RIM chip responds by asserting its WAIT and REQ outputs. (WAIT will return to its normal state near the end of the cycle. REQ is the output of a transparent latch gated by AS, and will, therefore, remain true until AS goes high with neither $\overline{\text{IOREQ}}$ nor $\overline{\text{MREQ}}$ asserted.) When the Address-Data bus arbitration logic grants the processor a cycle, the RIM chip asserts $\overline{\text{ADIE}}$ and $\overline{\text{AIE}}$ to gate address information from the processor onto the Address-Data bus. (The signal $\overline{\text{ADIE}}$ is used

to gate either address or data onto the Address-Data bus, while $\overline{AIE}$ is used to gate the higher order address bits.) On read cycles, determined by R/$\overline{W}$ being high at the trailing edge of AS, the RIM chip latches the address information, returns $\overline{ADIE}$ to its normal state, and places read data on the Address-Data bus. On write cycles, determined by R/$\overline{W}$ being low at the trailing edge of AS, the RIM chip latches the address information, and asserts $\overline{ILE}$ to gate processor data onto the Address-Data bus. The input $\overline{DWR}$ may be used, if necessary, to cause the RIM to wait until the processor data is ready. Some function of the REQ, WAIT, and $\overline{AIE}$ outputs, depending on the particular host processor and system bus structure, can be used to indicate the completion of the cycle.

The arbitration and bus control logic is designed so that the 2.2 microsecond limit will always be met, providing that the processor does not introduce excessive delays on the DWR (delayed write) control signal. To assure valid RIM operation, the delay between the fall of AS and the assertion of $\overline{DWR}$ must not exceed 950 ns. To assure the maximum I/O transfer rate (minimum number of processor wait cycles), the delay between the fall of AS and the assertion of $\overline{DWR}$ should not exceed 350 ns. If it can be determined that WRITE data will always be valid in time, then $\overline{DWR}$ should be grounded.

CONTROL/STATUS

The low order address bit determines the function. The higher order address bits and control signals are decoded to produce a chip select signal ($\overline{IOREQ}$) which is latched internal to the chip by AS. The interface functions include:

WRITE RIM COMMAND: issues a general purpose RIM control command. This command provides functions such as enable transmitter, disable receiver, clear selected status bits, etc.

READ STATUS: reads the eight-bit RIM status latch.

WRITE STATUS MASK: loads a mask register that enables interrupts from selected status bits.

BUFFER ACCESS

The low order ten address bits address the buffer. The higher order address bits and control signals are decoded to identify a processor access request command to the RAM buffer ($\overline{\text{MREQ}}$), which is latched internal to the chip by AS.

The buffer address is latched in the input latches. There are two input latches: the HIGH ADDR LATCH for A9-A8, and the ADDR/DATA IN LATCH for AD7-AD0. Separate enable signals are employed to allow the HIGH ADDR LATCH to supply A9-A8 continuously while the ADDR/DATA IN LATCH is disabled (e.g., during processor read operations).

During processor data transfers, the RIM chip acts as a memory control unit, interpreting $\overline{\text{MREQ}}$, R/$\overline{\text{W}}$, and $\overline{\text{DWR}}$, and generating control strobes for the buffer memory and the processor interface. The RIM chip will assert WAIT for as long as necessary. This halts the processor, allowing the RIM chip to coordinate the processor with its own memory cycle timing. The signal WAIT can also be used to latch the data on processor read cycles.

ANATOMY OF A HUB

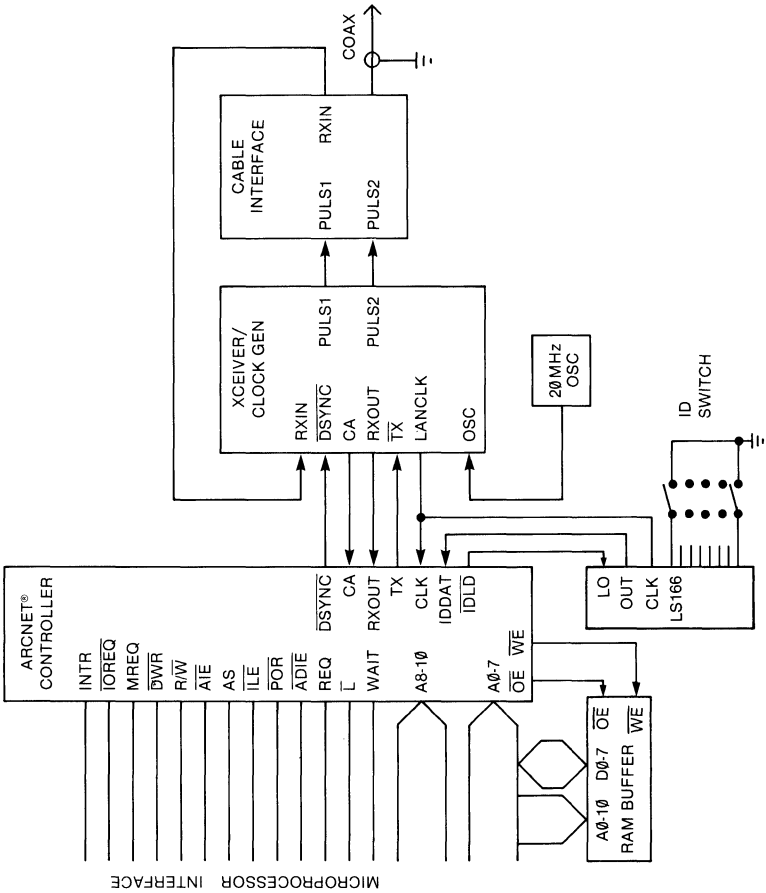
Each port of a hub consists of a line driver and receiver identical to those in the RIMs. The outputs of the receivers are OR'ed together. In its idle state, the hub has all of the receivers enabled. As soon as one of the receivers receives anything, the hub latches into a state in which the output of that receiver is fed to the inputs of the transmitters for all of the other ports. Meanwhile, all of the other receivers are disabled. The hub remains in this state until the transmission it is repeating is finished, and then drops back into its idle state.

The determination of when a transmission is finished is based on time. Since, there are never more than nine consecutive spacing elements in a transmission (the start element and eight zeros), a dipulse is received at least once every ten unit intervals, or 4.0 μs. The RIMs have a turnaround time of something greater than 12 μs; there are at least 12 μs between the end of the last data element of one transmission and the start of the alert burst of the next. Were it not for the reflection problem, the hub could drop back into its idle state when the receiver has not heard anything for some period of time between 4.0 and 12.0 μs.

To provide protection against reflections from unterminated lines, the hub should not fall into its idle state until the reflections cease. The STANDARD ARCNET INTERFACE limits individual runs of coax to 2000 feet of RG-62 coax, which has a velocity of propagation of 84%. Thus, the reflection from an unterminated, or shorted, cable will return in less than 4.9 μs. Changing the 4.0 μs limit mentioned above to 4.9 μs will therefore make the hub ignore reflections.

In summary, then, an N port hub is a device with N+1 stable states: the idle state, in which all N receivers are enabled; and N active states, one corresponding to each of the N receivers driving N-1 transmitters. The

transition from the idle state to an active state is caused by the reception of a dipulse by one of the receivers; the transition back to the idle state is caused by 8.5 (+/- 40%) μs of silence.



Rim Block Diagram

CHAPTER 4.
SOFTWARE INTERFACE TO THE RIM

Contents

PROCESSOR INTERFACE--CONTROL/STATUS

The processor exchanges control and status information with the RIM by reading and writing two registers in the RIM chip. The high order bits of the address are decoded outside the chip to produce a chip select signal. The low order bit specifies the address of a specific I/O register. The register functions are:

AD0	READ FUNCTION	WRITE FUNCTION
0	Status	Mask
1	(reserved)	RIM Command

A brief explanation of the I/O functions is given below.

READ STATUS

Execution of a read from RIM address 0 latches the current RIM status bits into the status latch and places the latched status on the data bus. The status register format is as follows:

BIT	STATUS	INDICATION
0	TA	Transmitter Available. Indicates that RIM transmitter is available for use; any previous ENABLE TRANSMIT command has completed.
1	TMA	Transmitted Message Acknowledged. Indicates that the message sent by the last ENABLE TRANSMIT command was acknowledged by the receiving RIM.

2	RECON	Reconfiguration Flag. Indicates that a SYSTEM RECONFIGURATION has occurred since the last time this bit was reset by a CLEAR FLAGS command. (The occurrence of a SYSTEM RECONFIGURATION is determined by the line being idle for at least 78 μs.)
3	TEST	Test Flag. This bit is intended for test and diagnostic purposes. It will be zero under any normal operating conditions.
4	POR	Power on Reset. Indicates that the RIM chip has undergone a power-on-reset since the last time this bit was reset by a CLEAR FLAGS command.
5	ETS1	Extended Time-out Status 1. Indicates the state of the ET1 input pin of the RIM chip. This pin is used to configure the RIM chip for special applications and will be high (i.e., the status bit will be one) under any normal operating conditions.
6	ETS2	Extended Time-out Status 2. Indicates the state of the ET2 input pin of the RIM chip. This pin is used to configure the RIM chip for special applications and will be high (i.e., the status bit will be one) under any normal operating conditions.
7	RI	Receiver Inhibited. Indicates that the RIM receiver is inhibited; no messages will be received until an ENABLE RECEIVE command is issued. If an ENABLE RECEIVE command has been issued since the last power on reset, RI indicates that a message has been received and is available in the buffer.

WRITE MASK

The RIM is capable of generating an interrupt signal to the processor when selected status bits become true. A write to the MASK register specifies the status bits that can generate an interrupt. A one bit in a position of the mask register enables an interrupt from the corresponding status bit. Status bits 1 (TMA), 5 (ETS1), and 6 (ETS2) will never cause interrupts--they have no mask bit. Status bit 4 (POR) is nonmaskable and will always cause an interrupt. The four maskable status bits are AND'ed with their respective mask bits, and the results, along with the POR status bit, are OR'ed to produce the processor interrupt request signal.

The interrupt request signal is negated when the interrupting status bit is reset, or when the corresponding bit in the mask register is set to zero.

WRITE RIM COMMAND

The processor issues commands to the RIM by writing a command byte to RIM I/O address 1, the RIM command register. The commands are:

COMMAND BYTE		FUNCTION
00 000	001	DISABLE TRANSMITTER - Causes transmitter to cancel an uncompleted transmit command. The next time the RIM is polled TA will be set. Note that failure of TA to come on within about 360 ms after issuing this command is an indication that there are no other RIMs connected to this one.

00 000	010	DISABLE RECEIVER - Causes receiver to cancel any uncomplete receive command. The next time the RIM is polled, RI will be set. Note that failure of RI to come on within about 360 ms after issuing this command is an indication that there are no other RIMs connected to this one.
00 0nn	011	ENABLE TRANSMIT from page nn - Clears TA and TMA status flags and commands the RIM to begin a transmit sequence, using buffer page nn, the next time it is polled. TA is set upon completion of the transmit sequence. TMA will have been set by this time if the RIM has received an acknowledgment from the destination RIM. (Note that this is strictly a hardware level acknowledgment which is sent by the receiving RIM before its host processor is even aware of the arrival of the message. Note also that the acknowledgment may get lost due to line errors, etc., so that TMA not being set is no guarantee that the message was not received.) This command should not be issued unless TA is true!
b0 0nn	100	ENABLE RECEIVE to page nn- Clears RI status flag and allows the RIM to receive messages into buffer page nn. If bit 'b' is a one, the RIM will accept broadcasts as well as messages addressed to it. If 'b' is zero, only messages

		addressed to this particular RIM will be accepted. RI is set upon successful reception of a message.
00 0rp	110	CLEAR FLAGS - Conditionally clears Power-On-Reset and RECON status bits. If p=1, it clears the Power-On-Reset status flag. If r=1, it clears system RECON status flag.

The first data byte of every ARCNET message is, by convention, a SYSTEM CODE. The SYSTEM CODE acts as a protocol identifier, allowing a number of systems using totally independent message formats to coexist on a single physical network. SYSTEM CODES for different operating systems, manufacturers, etc., are assigned through the DATAPOINT Product Marketing Department, Local Area Networking Group, 512/699-7151.

To transmit a message, the processor selects an inactive buffer (a buffer not specified by any uncompleted transmit or receive command) and writes into it in the following format:

ADR	CONTENTS	FUNCTION
0	xxx	Unused (RIM sends local ID regardless of this value)
1	DID	Destination IDentifier
2	CP	Continuation Pointer (buffer address of first data byte)
. . .		
CP	data	First data byte
. . .		
255	data	Last data byte (always at end of buffer)

The processor then waits for TA (status bit 0) to be true and gives the ENABLE TRANSMIT command, clearing both TA and TMA. At the completion of its transmit sequence, the adapter conditionally sets TMA (status bit 1) and then sets TA.

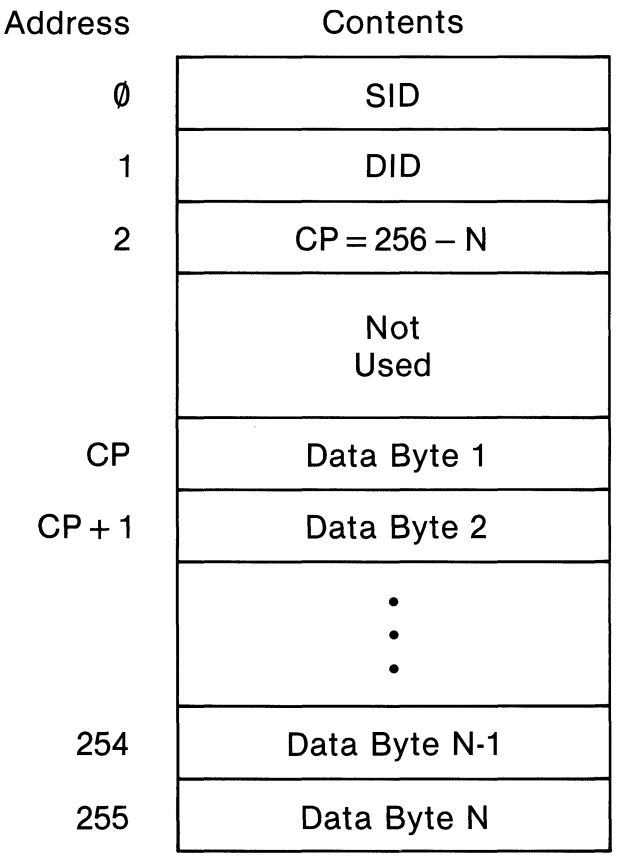
If the message is not a BROADCAST, the RIM will wait for a free buffer (RI false) at the destination before sending the packet. Note that if the host processor at the destination is not servicing its RIM, the adapter at the source will never find a free buffer

and never set TA. There must, therefore, be a software time-out on TA. When this timer times out, the processor should disable the TRANSMITTER to force the RIM to abandon the transmission. Note that if the DISABLE TRANSMITTER command does not cause TA to return true within 360 ms, it is an indication that there are no other RIMs in the system.

To enable the receiver, the processor selects an inactive buffer to use, waits for RI (status bit 7) to be true, and gives the ENABLE RECEIVE command, clearing RI. When a packet addressed to the local adapter or sent as a BROADCAST (if reception of BROADCASTs was enabled) is completely and correctly received and stored in the selected buffer, the adapter sets RI. The buffer contents are as follows:

ADR	CONTENTS	FUNCTION
0	SID	Source IDentifier
1	DID	Destination IDentifier (local ID or 0)
2	CP	Continuation Pointer (buffer address of first data byte)
. . .		
CP	data	First data byte
. . .		
255	data	Last data byte (always at end of buffer)

The format of the buffers is shown below.



N = Data Length
SID = Source ID (Not Used For Transmit Buffers)
DID = Destination ID (Ø For Broadcasts)

CHAPTER 5.
DIAGNOSTIC FEATURES OF THE RIM CHIP

Contents

CHIP TEST DIAGNOSTICS

The chip test diagnostics use the RIM chip input pins TEST1 and TEST2, in combination with input pins ET1 and ET2, to select from among a number of internal diagnostic functions. These functions are designed to simplify testing the RIM chip in a dedicated test system and normally cannot be used once the chip is incorporated into a PC board. The tests require intricate synchronized manipulation of the four test control pins, the two clocks, POR, etc., and are typically run on automated test equipment by the chip manufacturer.

BOARD TEST DIAGNOSTICS--ECHO TEST

The board test diagnostic function, the ECHO TEST, allows testing of the RIM chip at the PC board level. This function requires direct access to the RIM circuitry, but does not require that the chip be removed from the PC board.

The ECHO TEST is designed to execute a partial test on the RIM chip and its buffer without using the local processor. It is particularly useful in single board systems for identifying whether the RIM is functional when the condition of the processor is unknown.

Grounding the $\overline{\text{ECHO}}$ pin on the RIM chip effectively 'locks' both RI and TA false: the receiver is always ready to accept a message, and the transmitter will send a message every time it gets the token. If the receiver and transmitter are both dealing with the same page of the buffer, as they will be following a power-on-reset, then the RIM will continuously retransmit, or echo, any message it receives.

The ECHO TEST uses a known good RIM and a processor (or equivalent test fixture) to transmit a message to the RIM under test and then receive one of the echoes. Since the echoes will have the same DID as the original, and since both the RIM under test and the known good RIM must be able to receive messages containing this DID, the test message should be a broadcast. The test sequence is as follows:

1. Through RIM commands, disable the transmitter and the receiver on the known good RIM.

2. Ground the $\overline{\text{ECHO}}$ pin on the RIM under test. Momentarily assert $\overline{\text{POR}}$ to set the transmit and receive page registers to zero, and to set the broadcast reception enabled flag.

3. On the known good RIM, load the transmit buffer with a broadcast packet, and enable the transmitter. When TA becomes true, enable the receiver.

4. When RI becomes true, check the contents of the receive buffer for correct data.

WARNING: This test should never be performed in an operating system! The continuous stream of (echoed) broadcasts from the RIM under test will undoubtedly be a source of consternation to other nodes in the network!

UNIT TEST DIAGNOSTIC FEATURES

The unit test features allow partial testing of the RIM through processor commands.

RIM ID

At POR, the RIM chip microcode will execute two write cycles to the buffer. The data stored is:

Address	Data
0000	0321 (octal)
0001	RIM ID

The constant 0321 may be read by the processor to determine (with reasonable probability) that the test function is working. The processor may then read address 0001 to determine the RIM ID. This is a fairly powerful test in that it verifies at least partial operation of:

- ID logic (on-chip and off-chip)
- Major portions of the on-chip logic
- RIM-Write-RAM and Processor-Read-RAM cycles.

TEST FLAG

The TEST FLAG is controlled via a special command to the RIM chip:

00 00t 111	LOAD TEST FLAG - Sets or clears the TEST FLAG depending on whether t is one or zero, respectively.

The TEST FLAG provides a tool for checking the processor-to-RIM interface. Since the TEST FLAG is also a status bit, the effect of the RIM command may be verified by reading the status register. The interrupt structure may also be checked by enabling and disabling interrupts with the WRITE MASK command, while setting and resetting the TEST FLAG.

Due to the sharing of some internal functions of the RIM chip, the TEST FLAG should not be set unless status bit TA is true.

CHAPTER 6.
NONSTANDARD OPERATION OF THE RIM CHIP

Contents

NON-COMPATABILITY WARNING

The RIM chip is capable of being configured for two different types of 'nonstandard' operation. The maximum allowed propagation delay may be increased to provide for operation over large distances, and messages longer than 253 bytes may be sent.

WARNING: The use of these features results in an system that will be incompatible with standard ARCNET-based systems. Therefore, these features should only be implemented after careful consideration!

EXTENDED DISTANCE OPERATION

The Extended Time-out feature allows the RIM chip to operate over greater distances than standard RIMs. DC levels at the ET2 and ET1 inputs control the maximum distance over which the RIM can operate. The RIM's distance limitations are a function of internal time-outs which are part of the RIM protocol. The two types of internal time-outs are the Response Time-out and the RECON Time-out. The Response Time-out determines how long the RIM waits for a response after sending a message, and thus also determines the maximum propagation delay allowed between any two nodes. The RECON Time-out occurs when a RIM fails to receive an invitation to transmit from another RIM in the expected amount of time.

Altering the values of ET2 and ET1 cause the time-outs to change as follows:

ET2	ET1	MAX PROP DELAY (microseconds)	RECON TIME-OUT (seconds)	TIME TO RECONFIGURE (milliseconds)
1	1	31	0.84	24 to 61
1	0	131	1.68	77 to 223
0	1	271	1.68	149 to 439
0	0	549	1.68	291 to 603

WARNING: All RIMs on the network *must* use the same time-outs; the ARCNET standard is the 31 microsecond (ET1 = ET2 = VCC) setting.

EXTENDED LENGTH MESSAGE OPERATION

The extended length message feature allows the RIM chip to handle messages longer than the standard 253 bytes. Use of this feature involves both hardware and software considerations.

HARDWARE CONSIDERATIONS

RIMs supporting the extended length message feature require a 2k buffer, organized as four 512-byte pages, instead of the standard 1k buffer. RIM chip outputs A9 and A10 select the buffer page, as in the standard configuration, but are connected to system address lines A9 and A10 instead of A8 and A9. RIM chip output A8, which is unused in the standard configuration, provides the eleventh bit of address for the buffer.

SOFTWARE CONSIDERATIONS

Use of the extended length message feature is controlled via a special command to the RIM chip:

00 00c 101	DEFINE CONFIGURATION - The Long Packet Enable flag, an internal flag in the RIM chip, is set to the value of c. When this flag is set, a continuation pointer (contents of address 2 in a buffer page) value of zero indicates that the actual continuation pointer is to be found at address 3 and that the message ends at address 511 instead of address 255.

Messages of length 1 to 253 are sent in the usual manner, with the last byte of the message located at buffer address 255 and the contents of address 2 equal to the buffer address of the first byte of the message.

Messages of length 257 to 508 are loaded into the buffer with the last byte at address 511, the contents of address 2 set to zero, and the contents of address 3 equal to the buffer address of the first byte of the message.

NOTE: Messages of length 254 to 256 must be padded out to a length of at least 257 bytes in order to be handled.

COMPATIBILITY CONSIDERATIONS

RIMs equipped and configured for extended length message operation can coexist in the same system with standard RIMs. The DEFINE CONFIGURATION command merely informs the RIM chip of the existence of the 2k buffer (as opposed to the standard 1k) and thus should only be issued at initialization time. Operation with standard length messages proceeds in the normal fashion.

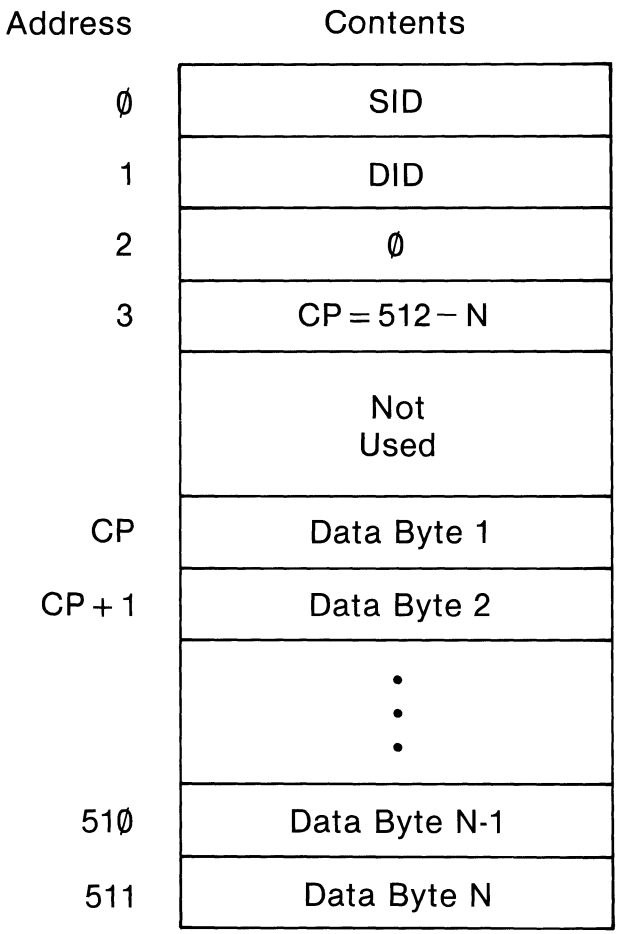
If an extended length message is sent to a RIM that does not have its Long Packet Enable flag set, the receiver will ignore it. The transmitting RIM will set TA, but not TMA.

If an attempt is made to send an extended length message when the DEFINE CONFIGURATION command has not been used to set the Long Packet Enable flag, the packet will not be sent and TA will stay off until a DISABLE TRANSMITTER command is issued. The host software sees this exactly as it sees the case of a message to a RIM that never makes a receive buffer available.

The final compatibility consideration involves compatibility with existing DATAPOINT ARCNET

systems using RIMs implemented without the RIM chip. To insure compatibility with such systems, messages longer than 320 bytes should not be sent.

The format of the extended length buffers is shown below.



N = Data Length
SID = Source ID (Not Used For Transmit Buffers)
DID = Destination ID (Ø For Broadcasts)

APPENDIX A.
DETAILED TIMING INFORMATION

Contents

DETAILED TIMING INFORMATION

The following information is provided for the benefit of users wishing to perform their own performance analysis. The lengths of the five types of transmission are shown below.

INVITATIONS TO TRANSMIT (ITT)

ALERT BURST	6 bits =	2.4 μs
EOT, DID, DID	3 x 11 = 33 bits =	13.2 μs
		15.6 μs

FREE BUFFER ENQUIRIES (FBE)

ALERT BURST	6 bits =	2.4 μs
ENQ, DID, DID	3 x 11 = 33 bits =	13.2 μs
		15.6 μs

PACKETS (PAC)

ALERT BURST	6 bits =	2.4 μs
SOH, SID, DID, DID,CP	5 x 11 = 55 bits =	22.0 μs
n characters	n x 11 = 11n bits =	4.4n μs
CRC, CRC	2 x 11 = 22 bits =	8.8 μs
		33.2 + 4.4n μs

ACKNOWLEDGMENTS (ACK)

ALERT BURST	6 bits =	2.4 μs
ACK	1 x 11 = 11 bits =	4.4 μs
		6.8 μs

NEGATIVE ACKNOWLEDGMENTS (NAK)

ALERT BURST	6 bits =	2.4 μs
NAK	1 x 11 = 11 bits =	4.4 μs
		6.8 μs

There are also a few important delay constants and propagation time definitions required for analysis. These are described below.

Turnaround Time (Tta) = 12.6 μs
The time from the end of a received transmission until the start of a response.

Token Propagation Time (Tpt)
The propagation time between the node holding the token and the node to which that node passes the token (i.e., the prop time related to the token).

Message Propagation Time (Tpm)
The propagation time between the node holding the token and the node to which that node sends a message (i.e., the prop time related to the message).

Broadcast Delay Time (Tbd) = 15.6 μs
The time from the end of a transmitted broadcast packet until the start of a token pass.

Response Time-out (Trp) = 74.6 μs
The maximum time to wait for a response: equal to twice the maximum allowable propagation delay plus Turnaround Time.

Recovery Time (Trc) = 3.4 μs
The time from the end of a Response Time-out until the start of a token pass.

Given the above numbers, it is possible to calculate the time that the token spends in the possession of any one node. A number of cases are detailed below. In each case, the time being calculated is the time from the start of one token pass to the start of the next token pass.

TOKEN PASS

ITT	15.6 μs
Tta	12.6 μs + Tpt

	28.2 μs + Tpt

TOKEN PASS and MESSAGE

ITT	15.6 μs
Tta	12.6 μs + Tpt
FBE	15.6 μs
Tta	12.6 μs + Tpm
ACK	6.8 μs
Tta	12.6 μs + Tpm
PAC3	3.2 μs + 4.4n μs
Tta	12.6 μs + Tpm
ACK	6.8 μs
Tta	12.6 μs + Tpm

	141.0 μs + 4.4n μs + Tpt + 4Tpm

TOKEN PASS and MESSAGE (receiver inhibited)

ITT	15.6 μs
Tta	12.6 μs + Tpt
FBE	15.6 μs
Tta	12.6 μs + Tpm
NAK	6.8 μs
Tta	12.6 μs + Tpm

	75.8 μs + Tpt + 2Tpm

TOKEN PASS and MESSAGE (broadcast)

ITT	15.6 μs
Tta	12.6 μs + Tpt
PAC	33.2 μs + 4.4n μs
Tbd	15.6 μs

	77.0 μs + 4.4n μs + Tpt

TOKEN PASS and MESSAGE (ACK gets lost)

ITT	15.6 μs
Tta	12.6 μs + Tpt
FBE	15.6 μs
Tta	12.6 μs + Tpm
ACK	6.8 μs
Tta	12.6 μs + Tpm
PAC	33.2 μs + 4.4n μs
Trp	74.6 μs
Trc	3.4 μs

	187.0 μs + 4.4n μs + Tpt + 2Tpm

TOKEN PASS and MESSAGE
(destination node does not exist)

ITT	15.6 μs
Tta	12.6 μs + Tpt
FBE	15.6 μs
Trp	74.6 μs
Trc	3.4 μs

	121.8 μs + Tpt

TOKEN PASS (no response)

ITT	15.6 μs
Trp	74.6 μs
Trc	3.4 μs

	93.6 μs

APPENDIX B.
TOOLS FOR ANALYZING SYSTEM PERFORMANCE

Contents

COUNTING RIMS

This section describes a technique which can be used to find the IDs of all the RIMs on a system. The technique is based on sending a dummy message to every possible RIM address. The system code 0200 octal (128 decimal) has been reserved for use with this scheme; therefore, a message with system code 0200 is ignored by all users of ARCNET. Note that the ARCNET hardware, the RIM, knows nothing about system codes. To the RIM, they are simply part of the data in a message. The responsibility for dealing with system codes, including ignoring any message with an 0200 system code, lies with the host software.

The state of the RIM with any given ID will fall into one of three categories:

- Receiver Enabled; the RIM with this address has its receiver enabled, ACKs free buffer enquiries, and accepts packets.

- Receiver Inhibited; the RIM with this address has its receiver inhibited and NAKs free buffer enquiries

- Nonexistent; the RIM with this address is disconnected from the system, turned off, or nonexistent.

Sending a message to RIMs in the first category, Receiver Enabled, will result in status bits TA and TMA both being set. This status is a guarantee of the existence of the RIM.

Sending a message to RIMs in the second category, Receiver Inhibited, will result in status bit TA staying off until forced on by a DISABLE TRANSMITTER command. TA staying off long enough for the token to visit every node is a guarantee that either the RIM

exists or the local RIM is not connected to any other RIMs! If the DISABLE TRANSMITTER command causes TA to come on within the maximum time it could take the token to visit every node, then the RIM in question exists. Otherwise, no RIMs, other than the local one, exist.

Sending a message to RIMs in the third category, Nonexistent, will result in status bit TA, but not bit TMA, being set. This, however, is not a guarantee of the RIM's nonexistence. The same result can be caused by the message, or the ACK, being damaged by noise, a RECONFIGURATION BURST, etc. The same result from some number of attempts, however, gives a reasonable level of assurance that there is no such RIM.

The outline of a procedure to determine the existence of a RIM with ID = N is shown below. (All numbers are in octal).

a) Prepare a RIM buffer with a one byte message to N. N (the DID) goes in location 1, 0377 (the CP) in location 2, and 0200 (the special System Code) in location 0377. Set a retry counter to 0.

b) Set a timer to 0 and issue the RIM an ENABLE TRANSMIT command. Wait for status bit TA to set or the timer to reach some limit. If TA sets, then go to step c; otherwise, (if the timer timed out) go to step e.

c) If status bit TMA is set, then go to step f; otherwise, go to step d.

d) If the retry counter has reached its limit, then go to step g. Otherwise, increment the retry counter and go to step b.

e) Set the timer to 0 and issue the RIM a DISABLE TRANSMITTER command. Wait for status bit TA to set or the timer to reach some limit. If TA sets, then go to step f; otherwise, (if the timer timed out) go to step h.

f) The RIM exists. The procedure terminates.

g) The RIM does not exist. The procedure terminates.

h) There are no other RIMs! The procedure terminates.

The time limit mentioned above should be long enough to insure that the local RIM receives the token. A limit of 360 ms is adequate even if there are 255 RIMs in the system and each of them sends a 253 byte message. Using a limit this large, however, could result in it taking over a minute and a half to find all the RIMs. A limit of 8 to 16 ms is adequate even if there are 255 RIMS in the system, as long as there are not too many messages being sent. The danger of the lower limit is that nonexistent RIMs will be mistaken for RIMs with their receivers inhibited. This results in a RIM count which is error on the high side. This count, however, forms an upper bound on the number of RIMs actually in the system. It is possible, therefore, to count the RIMs using a low limit on the time-out and then count them again using a limit based on the first count.

COUNTING MESSAGES

This section describes a technique that can be used to count the messages sent and keep track of the RIM that sent them. The technique is based on the fact that the RIM receiver, when enabled, writes the SID (Source ID) of every message it hears into location 0 of the current receive buffer, regardless of the DID (Destination ID) in the message. If the message is not directed to that RIM, nothing else is written into the buffer. It is impossible to tell to which RIM the message is addressed, or identify its contents, but it is possible to determine which RIM sent the message.

No status bits are changed when the SID is put in the buffer. The only way to detect that a message has been heard and ignored is to hang in a loop reading the buffer and waiting for it to change. The outline of a procedure to monitor traffic is described below.

a) Write a 0 into location 0 of a RIM buffer and issue the RIM an ENABLE RECEIVE command.

b) Test the byte at location 0 of the RIM buffer. If it is still 0, go to step b; otherwise, go to step c.

c) Record a message as having been sent by the RIM whose ID was read in step b. Go to step a.

Combined with a timer and some simple video pyrotechnics, this technique can be used to produce a very interesting monitor program which displays number of messages per second, total number of messages, etc.

TIMING THE TOKEN

This section describes a technique that can be used to measure the time it takes the token to visit every RIM in the system. The technique is based on the fact that when a DISABLE TRANSMITTER command is issued to the RIM, status bit TA will be set the next time the token is received. Thus, a loop that issues an ENABLE TRANSMIT command immediately followed by a DISABLE TRANSMITTER command can time the token. The outline of a token timing procedure is shown below. (All numbers are in octal).

a) Prepare a RIM buffer with a one byte message to the local RIM. MYID (the local RIM ID) goes in location 1, 0377 (the CP) in location 2, and 0200 (the special System Code for test messages) in location 0377.

b) Issue an ENABLE TRANSMIT command immediately followed by a DISABLE TRANSMITTER command. Wait for status bit TA to come on. Set a timer to 0. Issue an ENABLE TRANSMIT command immediately followed by a DISABLE TRANSMITTER command.

c) Wait for status bit TA to come on. Record the value of the timer. Issue an ENABLE TRANSMIT command immediately followed by a DISABLE TRANSMITTER command.

d) Process the recorded timer value.

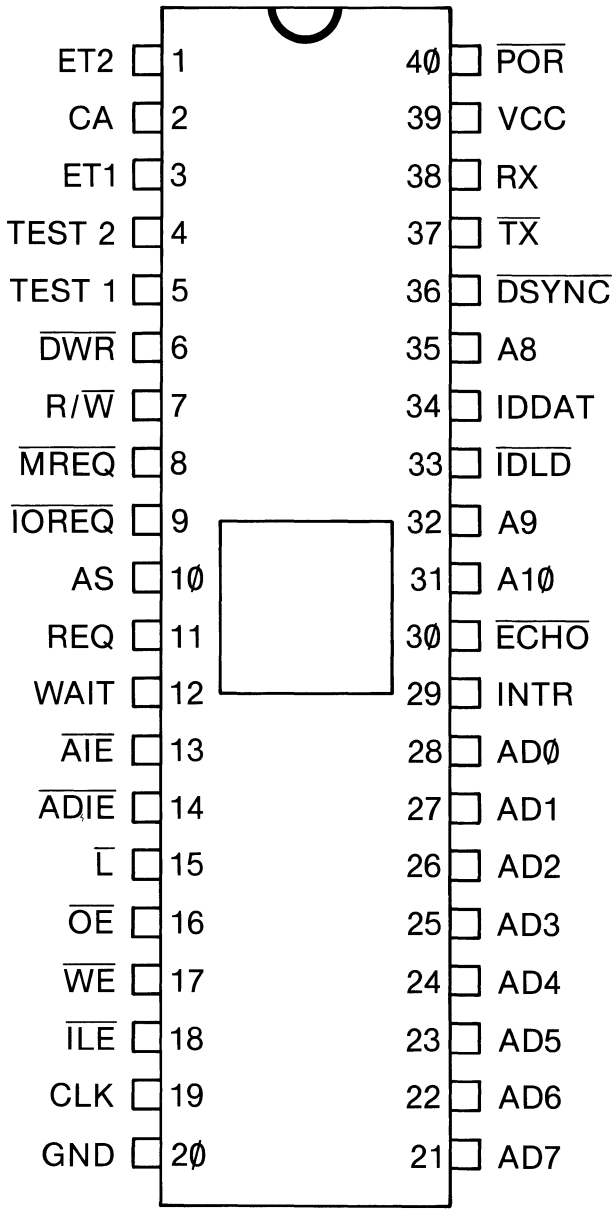
e) Test status bit TA. If it is set, then go to step b; otherwise, go to step c.

Step b synchronizes the program with the token by waiting for the token to be received before starting the timer. The first ENABLE TRANSMIT command in step b is the only one in the procedure that may actually result in a message being sent. Thus, if the processing in step d is fast enough for the program to stay in sync with the token, the program is a completely passive monitor: it measures the token time without affecting it in any way. The test in step e checks to see if the token has already come around again. If it has, it is necessary to return to step b to get back in sync.

APPENDIX C.
RIM CHIP PINOUT AND PIN DESCRIPTION

Contents

PIN CONFIGURATION



SYMBOLS AND FUNCTIONS

SYMBOL	FUNCTION
A10-A9 (Output, tri-state)	PAGE bus: The two most significant bits of the buffer memory address These pins are system address bits 9-8 in a 1K buffer system, and bits 10-9 in a 2K buffer system.
A8 (Output, tri-state)	ADDRESS: Used as an eleventh address bit in 2K buffer systems. This pin is unused in a 1K buffer system, and is system address bit 8 in a 2K buffer system.
AD7-AD0 (Input/ Output, tri-state)	Address-Data bus: The lower eight bits of the buffer memory address and the eight bit data path in and out of the RIM chip. AD0 is also used for I/O command decoding of processor control/status commands to the RIM.
$\overline{\text{IOREQ}}$ (Input)	I/O REQUEST: Driven low by the processor to request use of the RIM bus to transfer control/status information. $\overline{\text{IOREQ}}$ is sampled by the RIM on the falling edge of AS.
$\overline{\text{MREQ}}$ (Input)	MEMORY REQUEST: Driven low by the processor to request use of the RIM bus to transfer data to or from the RAM buffer memory. $\overline{\text{MREQ}}$ is sampled by the RIM on the falling edge of AS.

SYMBOL	FUNCTION
R/$\overline{W}$ (Input)	READ/WRITE: Used by the processor to identify an access request (I/O or MEMORY) as either a read or a write cycle. R/$\overline{W}$ is sampled by the RIM on the falling edge of AS.
AS (Input)	ADDRESS STROBE: Used by the RIM to sample the status of $\overline{IOREQ}$, $\overline{MREQ}$, and R/$\overline{W}$. RIM bus arbitration is initiated on the falling edge of AS if either $\overline{IOREQ}$ or $\overline{MREQ}$ is asserted.
REQ (Output)	REQUEST: Acknowledgment from the RIM that a processor I/O cycle or memory cycle request has been sampled.
WAIT (Output)	WAIT: Asserted by the RIM chip at the start of processor access cycles to indicate that it is not ready to transfer data. WAIT is negated when the RIM is ready for the processor to complete the cycle.
$\overline{DWR}$ (Input)	DELAYED WRITE: Used by the processor during processor write cycles to inform the RIM that there is valid data on the data bus. $\overline{DWR}$ may be asserted asynchronously, and is not used on read cycles. If the processor will always meet setup for WRITE data, $\overline{DWR}$ should simply be grounded.

SYMBOL	FUNCTION
INTR (Output)	INTERRUPT: Asserted by the RIM to indicate that an enabled interrupt condition has occurred. INTR is negated when the interrupting status condition is reset, or when the corresponding mask bit is set to zero.
$\overline{\text{ILE}}$ (Output)	INTERFACE LATCH ENABLE: An active low signal that gates the processor output data onto the Address-Data bus during processor write cycles.
$\overline{\text{ADIE}}$ (Output)	ADDRESS DATA INPUT ENABLE: An active low signal that enables the processor to drive the Address-Data bus.
$\overline{\text{AIE}}$ (Output)	ADDRESS INPUT ENABLE: An active low signal that enables the processor to drive A10-A8.
$\overline{\text{L}}$ (Output)	LATCH: An active low signal that latches the lower eight bits of the Address-Data bus into an eight bit address latch external to the RAM.
$\overline{\text{WE}}$ (Output)	WRITE ENABLE: Used to strobe data into the buffer memory. The trailing edge of WRITE ENABLE signals the RAM to execute a write operation.
$\overline{\text{OE}}$ (Output)	OUTPUT ENABLE: An active low signal that enables the RAM output data onto the Address-Data bus.

SYMBOL	FUNCTION
$\overline{\text{IDLD}}$ (Output)	ID LOAD: An active low signal that synchronously loads the setting of the ID switches into the external ID shift register.
IDDAT (Input)	ID DATA IN: An input that accepts the serialized ID switch setting. The ID is shifted into the RIM, high true, most significant bit first.
ET2-ET1 (Input)	EXTENDED TIME-OUT FUNCTION: See notes 1 and 4.
$\overline{\text{TX}}$ (Output)	TRANSMIT DATA: Serial transmit data output to the LINK transmitter.
RX (Input)	RECEIVE: Serial receive data input from the LINK receiver.
TEST1-TEST2 (Input)	TEST PINS 1 and 2: See notes 2 and 3.
$\overline{\text{ECHO}}$ (Input)	ECHO DIAGNOSTIC ENABLE: See notes 2 and 4.

SYMBOL	FUNCTION
CLK (Input)	CLOCK: A continuous 5MHz clock used for timing of the RIM bus cycles and for bus arbitration, serial ID input, and the RECON TIMER. The chip requires a high level drive on the CLK input.
CA (Input)	CA: A 5 MHz clock used to control the operation of the RIM sequencer. CA is periodically halted in the high state by $\overline{\text{DSYNC}}$ and restarted by RX. The chip requires a high level drive on the CA input.
$\overline{\text{DSYNC}}$ (Output)	DELAYED SYNC: Asserted by the RIM chip to cause the external clock generator logic to halt the CA clock.
$\overline{\text{POR}}$ (Input)	POWER ON RESET: Sets the program counter to zero and initializes various internal control flags and status bits. Sets the POR status bit, causing INTR to be asserted.
VCC	+5 Volt Supply.
GND	Ground.

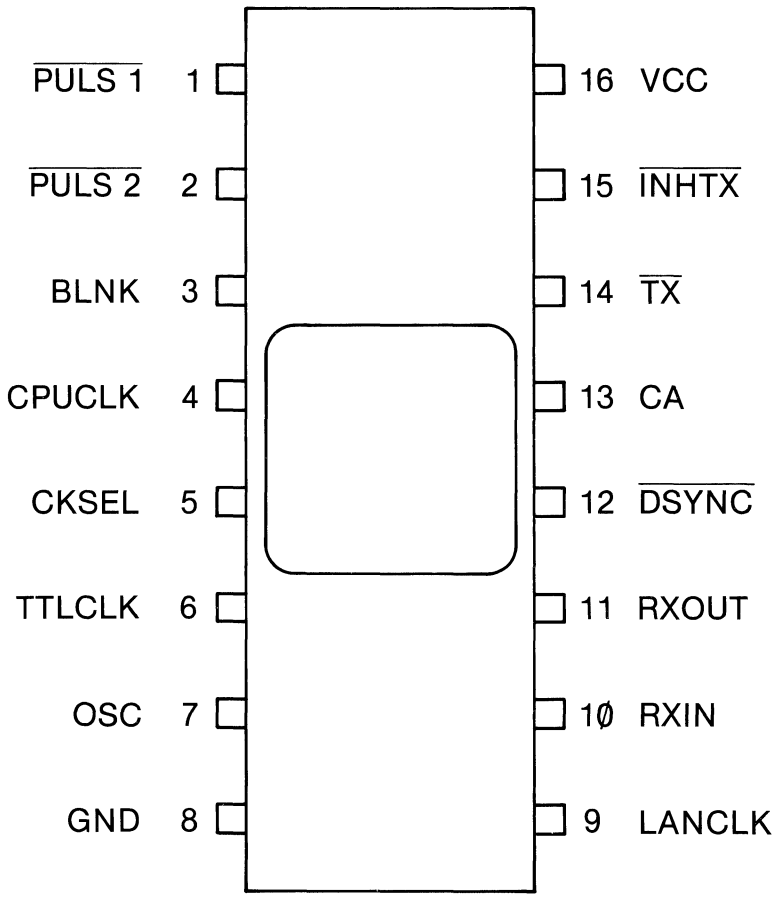
NOTES

NOTE	EXPLANATION
1	These pins are used to configure the RIM chip for special applications.
2	These pins are used only for testing.
3	These pins should be grounded for normal operation.
4	These pins should be tied high for normal operation.

APPENDIX D.
TRANSCEIVER AND CLOCK GENERATOR CHIP PINOUT AND PIN DESCRIPTION

Contents

PIN CONFIGURATION



SYMBOLS AND FUNCTIONS

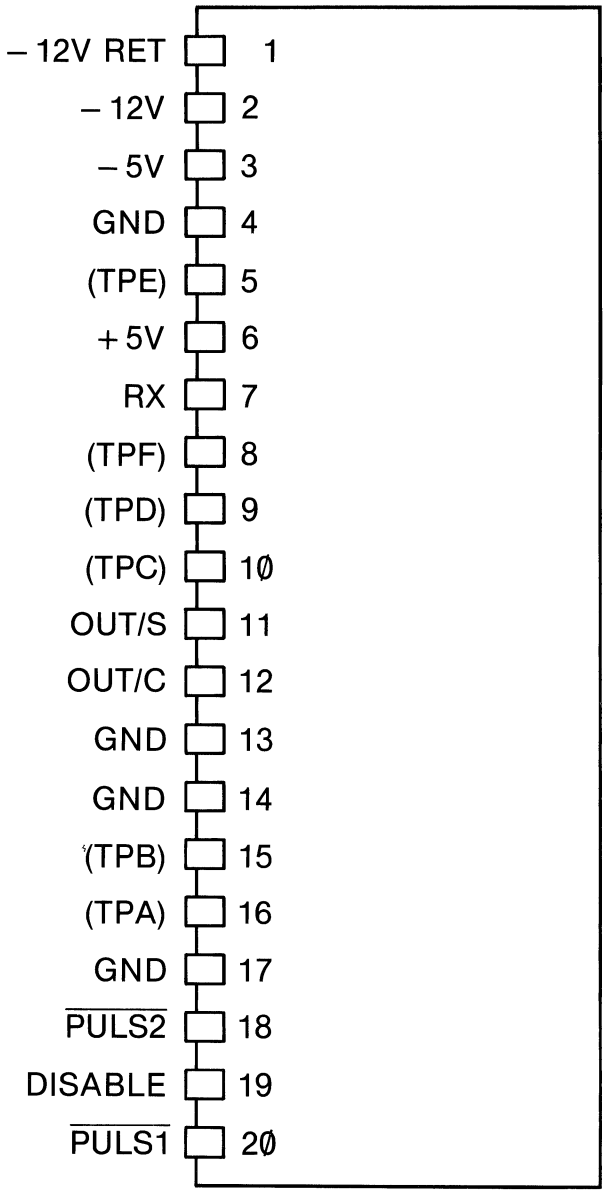
SYMBOL	FUNCTION
OSC (input)	Input for 20 MHz clock.
CKSEL (input)	Selects clock options available at TTLCLK and CPUCLK. When CKSEL is high, a 4 MHz clock is available at outputs CPUCLK and TTLCLK. When CKSEL is low, CPUCLK is the inverse of the signal input at TTLCLK.
LANCLK (output)	Free-running 5 MHz clock for controller.
CA (output)	5MHz clock which clocks data into the controller.
TTLCLK (input/output)	See CKSEL above.
CPUCLK (output)	See CKSEL above.
$\overline{\text{TX}}$ (input)	Serial data input from the controller.
$\overline{\text{PULS1}}$ (output)	Negative going output pulse which generates first half of output dipulse.
$\overline{\text{PULS2}}$ (output)	Negative going output pulse which generates second half of output dipulse.
RXIN (input)	Received data from the network.

SYMBOL	FUNCTION
$\overline{\text{DSYNC}}$ (input)	Input which when asserted halts clock CA.
RXOUT (output)	Received data to the controller.
$\overline{\text{INHTX}}$ (input)	Inhibits transmitting by forcing PULS1 and PULS2 high and BLNK low. Should be used at power on.
BLNK	Not used for normal operation.
VCC	+5 volt supply.
GND	Circuit ground.

APPENDIX E.
LINE DRIVER MODULE PINOUT AND PIN DESCRIPTION

Contents

PIN CONFIGURATION



NAMES AND FUNCTIONS

NAME	FUNCTION
$\overline{\text{PULS1}}$ (input)	Transmit PULSE 1 from receiver.
$\overline{\text{PULS2}}$ (input)	Transmit PULSE 2 from transceiver.
DISABLE (input)	Transmitter DISABLE function (note 3).
OUT/C (input/output)	To center conductor of coax.
OUT/S (input/output)	To shield of coax.
RX (output)	ARCNET received pulse to transceiver.
+5V	+5 Volt Supply (note 1)
-5V	-5 Volt Supply (note 1)
-12V	-12 Volt Supply (note 1)
-12V RET	-12 Volt Return (note 1)
GND	Circuit Ground (pins 4 & 17)
GND	Optional: Ground or N.C. (pins 13 & 14)
(TPA)	Test Point A (note 2)

NAME	FUNCTION
(TPB)	Test Point B (note 2)
(TPC)	Test Point C (note 2)
(TPD)	Test Point D (note 2)
(TPE)	Test Point E (note 2)
(TPF)	Test Point F (note 2)

NOTES

NOTE	EXPLANATION
1	The line driver module is designed to be powered from +5 volts and either -5 volts or -12 volts. Unused pins are left unconnected.
2	The maximum number of pins has been provided on the hybrid microcircuit to provide mechanical security with polarization. Six of these pins have been reserved for optional circuit test points. No electrical connections are made to these six pins in standard ARCNET applications.
3	The transmitter disable function (pin 19) is normally tied to GROUND (i.e., it is enabled) for most ARCNET applications. When used in the Active Hub, external circuitry uses this pin to disable the transmitter on the receiving channel.


DATAPOINT