SMSC™
STANDARD MICROSYSTEMS CORPORATION

COM20051

Integrated Microcontroller and ARCNET (ANSI 878.1) Interface

FEATURES

- High Performance/Low Cost
- Microcontroller Based on Popular 8051 Architecture
- Intel *8051 Instruction Set Compatible*
- Drop-In Replacement for 80C32 PLCC
- Network Supports up to 255 Nodes
- Powerful Network Diagnostics
- Maximum 507 Byte Packets
- Duplicate Node ID Detection
- Self-Configuring Network Protocol
- Retains all 8051 Peripherals Including Serial I/O and Two Timers
- Utilizes ARCNET Token Bus Network Engine
- Requires No Special Emulators
- 5 Mbps to 156 Kbps Network Data Rate
- Network Interface Supports RS-485, Twisted Pair, Coaxial, and Fiber Optic Interfaces
- Receive All Mode Allows Any Packet to Be Received

GENERAL DESCRIPTION

The COM20051 is a low-cost, highly-integrated microcontroller incorporating a high-performance network controller based on the ARCNET Token Bus Standard (ANSI 878.1). The COM20051 is based around the popular Intel 8051 architecture. The device is implemented using a microcontroller core compatible with the Intel 80C32 ROMless version of the 8051 architecture. The COM20051 is ideal for distributed control networking applications such as those found in industrial/machine controls, building/factory automation, consumer products, instrumentation and automobiles.

The COM20051 contains many features that are beneficial for embedded control applications.

The microcontroller is a fully-functional 16MHz 80C32 that is comparable to the Intel 80C32 with 2 timers. In contrast to other embedded controller/networking solutions, the COM20051 adds a fully-featured, robust, powerful, and simple network interface while retaining all of the basic 8051 peripherals, such as the serial port and counter/timers.

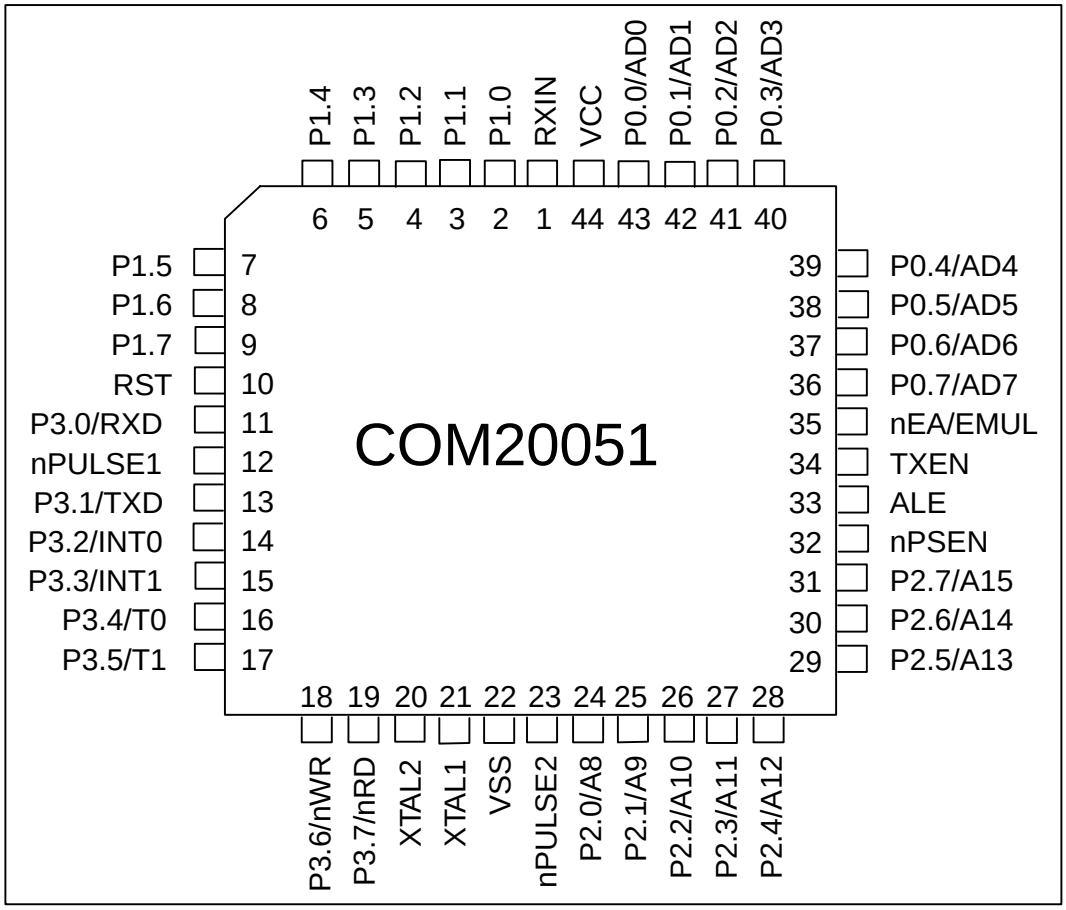
In addition, the COM20051 supports an Emulation Mode that permits the use of a standard 80C32 emulator in conjunction with the COM20051 to develop software drivers for the network core. *ARCNET core is mapped to a 256-byte page of the External Data Memory Space of the 80C32. This*

TABLE OF CONTENTS

80 Arkay Drive
Hauppauge, NY 11788
(631) 435-6000
FAX (631) 273-3123

provides for an easy interface between the CPU and the ARCNET core. The networking core is based around an ARCNET Token Bus protocol engine that provides highly-reliable and fault tolerant message delivery at data rates ranging from 5Mbps down to 156 Kbps with message sizes varying from 1 to 508 bytes. The ARCNET protocol offers a simple, standardized, and easily-understood networking solution for any application. The network interface supports several media interfaces, including RS-485, coaxial, and twisted pair in either bus or star topologies. The network interface incorporates powerful diagnostic features for network management and fault isolation. These include duplicate node ID detection, reconfiguration detection, receive all (monitor) mode, receiver activity, and token detection.

PIN CONFIGURATION



OVERVIEW

The COM20051 is essentially a network board-in-a-chip. It takes an 80C32 microcontroller core and an ARCNET controller and integrates them into a single device. ARCNET is a token passing-based protocol that combines powerful flow control, error detection, and diagnostic capabilities to deliver fast and reliable messages. The COM20051 supports a variety of data rates (5 Mbps to 156 Kbps), topologies (bus, star, tree), and media types (RS-485, coax, twisted pair, fiber optic, and powerline) to suit any type of application.

The ARCNET network core of the COM20051 contains many features that make network development simple and easy to comprehend. Diagnostic features, such as Receive All, Duplicate ID Detection, Reconfiguration Detection, Token, and Receiver Detection, all combine to make the COM20051 simple to use and to implement in any environment. The ARCNET protocol itself is relatively simple to understand and very flexible. A wide variety of support products are available to assist in network development, such as software drivers, line drivers, boards, and development kits. The COM20051 implements a full-featured 16MHz, Intel-compatible 80C32 microcontroller with all of the standard peripheral functions, including a full duplex serial port, two timer/counters, one 8-bit general purpose digital I/O port, and interrupt controller. The 8051 architecture has long been a standard in the embedded control industry for low-level data acquisition and control. ARCNET and the 8051 form a simple solution for many of today's and tomorrow's low-level networking solutions.

In addition to the 80C32 and the ARCNET network core, the COM20051 contains all the address decoding and interrupt routing logic to interface the network core to the 80C32 core. The integrated 8051/ARCNET combination provides an extremely cost-effective and space-efficient solution for industrial networking applications. The COM20051 can be used in a stand-alone embedded application, executing control algorithms or performing data acquisition and communicating data in a master/slave or *peer-to-peer* configuration, or used as a slave processor handling communication tasks in a multi-processing system.

DESCRIPTION OF PIN FUNCTIONS

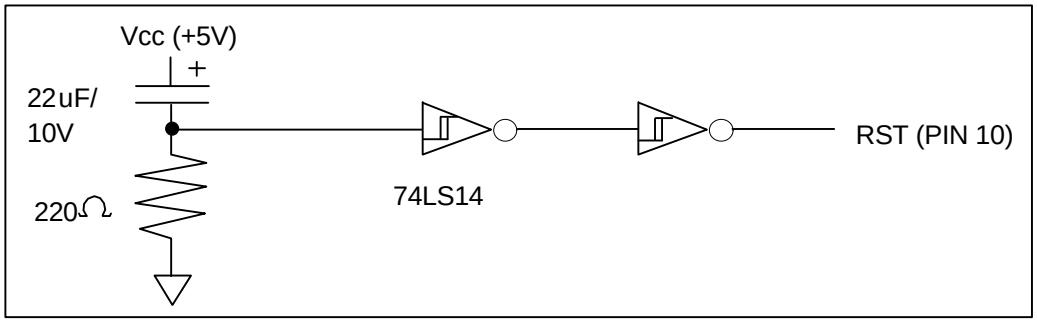
PIN NO.	NAME	SYMBOL	DESCRIPTION
1	Receive In	RXIN	Input. Network receiver input.
2-9	P1.0-1.7	P1.0-1.7	Input/Output. Port 1 of the 8051. General purpose digital I/O port.
10	Reset	RESET	Input. Active high reset.
11	P3.0	P3.0	Input/Output. Port 3 bit 0 of the 8051. RX input of serial port.
12	nPulse 1	nPULSE1	Output. Network output. Open-drain when backplane mode is invoked, otherwise it is a push-pull output.
13-19	P3.1-3.7	P3.1-3.7	Input/Output. Port 3 bits 1-7 of the 8051.
20, 21	Crystal Oscillator	XTAL1, XTAL2	Input. Oscillator inputs 1 and 2.
22	Ground	VSS	Ground pin.

DESCRIPTION OF PIN FUNCTIONS

PIN NO.	NAME	SYMBOL	DESCRIPTION
23	nPulse 2	nPULSE2	Output. Network output. Outputs a synchronous clock at 2x the data rate when backplane mode is invoked.
24-31	P2.0-2.7	P2.0-2.7	Input/Output. Port 2 of the 8051. High order address bus.
32	nProgram Store Enable	nPSEN	Output.
33	Address Latch Enable	ALE	Output.
34	Transmit Enable	TXEN	Output. This signal is used to enable the drivers for transmitting. The polarity of this signal is programmable by grounding the nPULSE2 pin prior to the POWER-UP. nPULSE2 floating prior to the power-up = TXEN active high nPULSE2 grounded prior to the power-up = TXEN active low. (This option is available only in the Backplane mode).
35	nEnable	nEA	Input. When high, causes the 8051's outputs to tri-state. When low, allows the 8051 to address external memory. Must be low to execute code from the embedded 8051.
36-43	P0.7-0.0	P0.7-0.0	Input/Output. Port 0 of the 8051. Multiplexed low order address/data bus.
44	Power Supply	VCC	+5V power supply.

RESET CIRCUIT FOR THE COM20051

The power on reset circuit for the COM20051 should be designed to provide a clean, fast transition time TTL input to the COM20051. Sufficient signal high time on RST (pin 10) should be provided after Vcc reaches +5V DC. The following circuit, which provides an 8ms power-on reset pulse, is recommended:



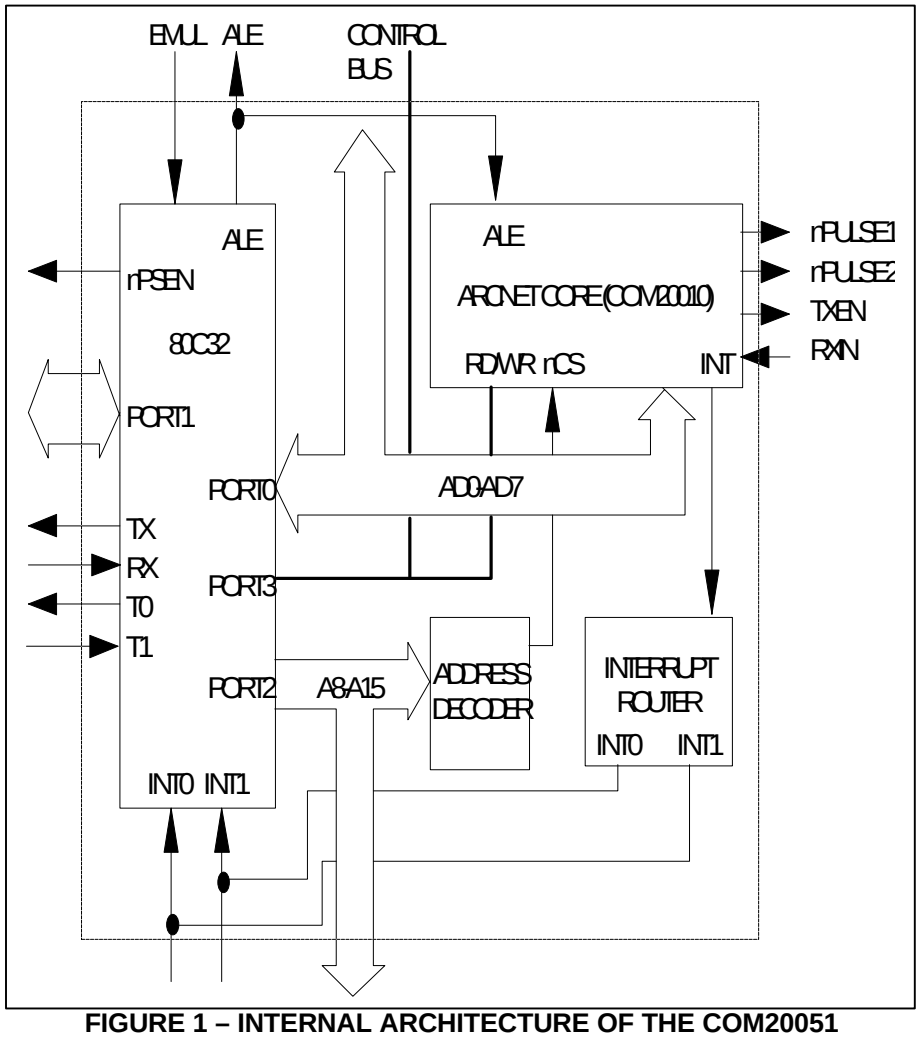


FIGURE 1 – INTERNAL ARCHITECTURE OF THE COM20051

BASIC ARCHITECTURE

The COM20051 consists of four functional blocks: the 80C32 microcontroller core, ARCNET network cell (includes 1K of buffer RAM), programmable address decoder, and programmable interrupt router. The internal architecture of the COM20051 is shown in Figure 1.

The 80C32 microcontroller is a full ROMless implementation of the popular Intel 8051 series. The ARCNET network core is similar in architecture to SMSC's popular COM20020 family of ARCNET controllers and retains the same command and status flags of previous ARCNET controllers. The programmable address decoder maps the ARCNET registers into a 256-byte page anywhere within the External Data Memory space of the 80C32. The ARCNET core was mapped to the External Data Memory space to simplify software and application development and for production test purposes. *ARCNET core is available to the developer when working with the 8051 emulator.* When the COM20051 is put into Emulate mode, the internal microcontroller is put into a high impedance state, thus allowing an external In-Circuit Emulator (ICE) to program the ARCNET core. The advantage of this approach versus mapping the ARCNET registers into the internal memory (Special Function) area of the 80C32 is that dedicated software development tools will not be necessary to debug application software. Since a majority of 8051 applications use only a small portion of the Data Memory space, there is no penalty paid for used address space. There will also be no penalty in execution time, since cycle times for external data memory accesses and internal direct memory moves are identical. The network interrupt can be routed to either of the two external interrupt ports or can be assigned to one of the general purpose I/O ports. The ARCNET interrupt is internally wire ORed with the external interrupt pin to allow greater system flexibility.

80C32 ARCHITECTURE AND INSTRUCTION SET

The 80C32 microcontroller core is identical to the 16MHz Intel 80C32 in all respects *except for the absence of Timer 2.* Please refer to the Intel Embedded Microcontrollers and Processors Databook, Volume 1, for details regarding the 8051 architecture, peripherals, instruction set, and programming guide. Note that any access to the internal ARCNET core or any external memorry access is *visible* on the pins of the COM20051.

The following differences apply to the COM20051:

1. Oscillator frequency is 40MHz instead of 16MHz. This is necessary to derive a 20MHz clock for the ARCNET core. The processor still operates at 16MHz.

2. nEA pin - This pin must be tied to ground for normal internal processor operation. When tied to VCC, the COM20051 will enter the Emulate mode.

3. Unused pins - The COM20051 is packaged in a 44-pin PLCC. Network I/O is generated on the four unused pins of the standard 80C32 PLCC package. No DIP package is available.

4. Power Down operation - The Power Down mode can only be used in conjunction when the internal oscillator is being used. If an external oscillator is used and the Power Down mode is invoked, damage may result to the oscillator and to the COM20051.

Clock Speed

The COM20051 processor operates at 16MHz and the network controller at a maximum 40MHz clock rate. A single crystal oscillator is used to supply the two clocks: a 16MHz processor clock and a 20MHz network clock for the nominal 2.5 Mbps data rate. Pins 20 and 21 are designated as crystal inputs. When clocking with an external

oscillator, pin 21 (XTAL1) functions as the clock input.

Emulate Mode

The COM20051 contains a unique feature called the Emulate mode that most 8051-based peripheral devices do not accommodate. TheEmulate mode permits developers to access and program the internal ARCNET core using a standard low-cost 8032 emulator. This feature eliminates the need for expensive dedicated development equipment needed for other types of 8051-based peripheral devices. The Emulate mode is invoked by connecting the nEA pin to VCC. This causes the internal 80C32 processor to enter a HI-Z state and changes the state of the COM20051 pins according to the following table:

Table 1 - Emulate Mode

SIGNAL NAME	EMUL = 0	EMUL = 1
PORT 0	Bidirectional	Bidirectional
PORT 1	Bidirectional	HI-Z (except for pins designated as interrupt destinations)
PORT 2	Output	Input
INT0,1 (P3.2, P3.3)	Input	Output
RD/WR (P3.6, P3.7)	Output	Input
ALE	Output	Input
TX,T0, T1 (P3.1,3.4,3.5) nPSEN	Output	HI-Z

Address Decoding

The COM20051, as described previously, maps the ARCNET registers into the 80C32's External Data Memory space. This provides system flexibility because the location of the ARCNET registers can be located anywhere within the 64K External Data Memory space. The precise location can be resolved with a 256-byte page. The location of that page in the External Data Memory space is pointed to by the *read/write* Address Decode Register, as shown in Figure 2. The Address Decode Register is located at FFFFh of the External Data Memory space. It holds the upper 8 bits of the 16-bit address at which the 256 page boundary will start. This register must be programmed prior to any access to the ARCNET core. The default value is 0000h.

The ARCNET core register page must be mapped away from the external RAM prior to any access to the external RAM by the software. Failure to do this may result in incorrect data write and read operations to the external RAM as well as the core registers.

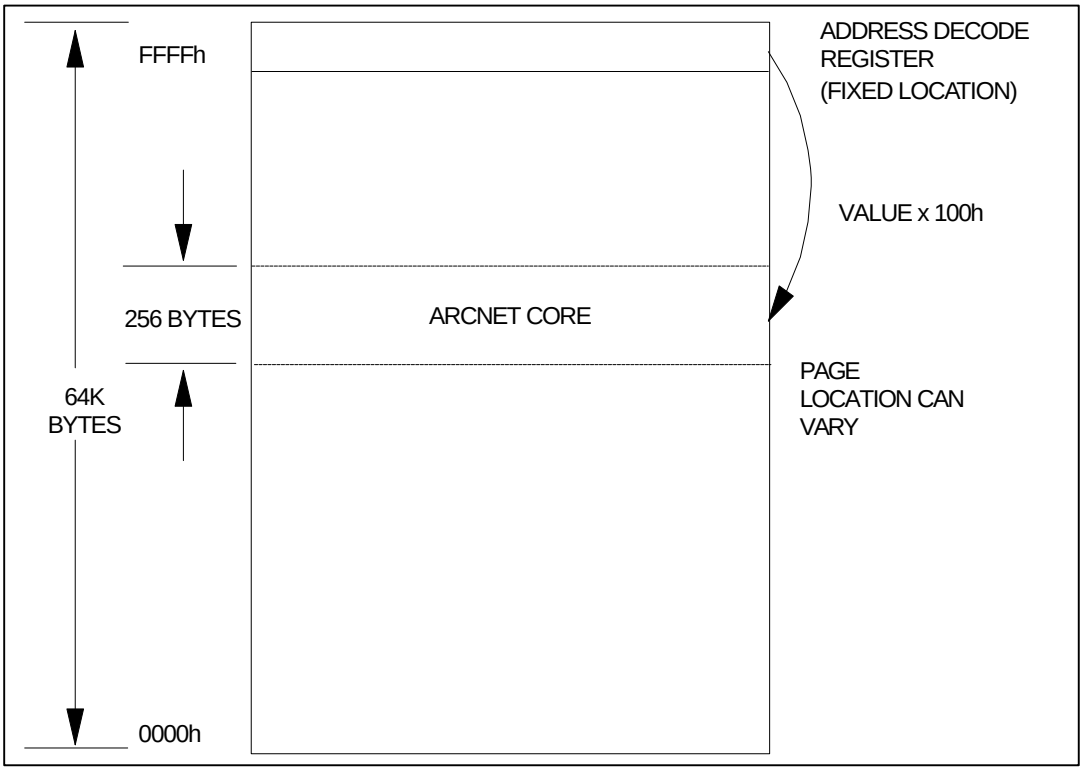


FIGURE 2 – COM20051 EXTERNAL DATA ADDRESS SPACE

ADDRESS DECODE REGISTER

NAME	*BIT 7*	*BIT 6*	*BIT 5*	*BIT 4*	*BIT 3*	*BIT 2*	*BIT 1*	*BIT 0*
ADR DEC	*A15*	*A14*	*A13*	*A12*	*A11*	*A10*	*A9*	*A8*

LOCATION: FFFFh of the External Data Memory space. Default: 00h

EXAMPLE: Address Decode Register = 80h
ARCNET core registers will be located at 8000h + Register offset (e.g. ARCNET Configuration Register offset = 06h, physical address = 8006h).

COM20051 MEMORY MAPPING

The COM20051 maps the Arcnet core into a 256 byte page of data memory space. This memory is physically located internally to the device and itrquote s default base address on power up is 0000h. This 256 byte page can be logically located anywhere within the 64K external data memory space while physically remaining on board. The location of this 256 byte page is pointed to by the Address Decode Register in the device. This Address Decode Register holds the upper 8 bits of the 16 bit address at which the 256 byte page boundary will start. The address of this Address Decode Register is FFFFh. This register is also logically located in external data memory space but physically located on the device. This register must be written to on power-up to properly locate the Arcnet core. .

The user must ensure that the Arcnet corerquote s 256 byte page does not conflict with external memory, otherwise data bus contention will result. As an example, if the user has 32K of external data memory located from 0000h to 7FFFh then the Arcnet core should be mapped above this area, 8000H is suggested. The user will write 80h to address FFFFh on power up to properly map the core to this location.

ARCNET Network Core - Overview and Architecture

ARCNET is a baseband token passing network protocol (ANSI 878.1). ARCNET features deterministic behavior, hardware-based network configuration, flexible topologies, several data rates, and multiple media support. Data rates varying from 5 Mbps to 156 Kbps and message sizes from 1 to 508 bytes are supported. Supported media includes RS-485, twisted pair, coax, fiber optic, and powerline in bus, star or tree topologies. ARCNET has enjoyed widespread use in the industrial community, finding a home in such applications as I/O control/acquisition, multi-processor communications, point-of-sale terminals, in-vehicle navigation systems, data acquisition systems, remote sensing, avionics, machine control, embedded computing, building automation, robotics, consumer products, and security systems.

The ARCNET core used in the COM20051 is similar in architecture to SMSC's 200XX series of Industrial ARCNET Controllers. The ARCNET core of the COM20051 contains a 1K x 8 internal RAM for packet buffering, Duplicate ID Detection, Receive All Mode, New Next ID Indicator, Excessive NACK Interrupt, Programmable Data Rates, Backplane Mode, Programmable Transmitter Enable, Polarity Receive Activity, Reconfiguration, Token Seen Indicators, and Network Mapping hooks. The ARCNET core of the COM20051 uses a software-programmable node ID, *enabling the user to program the Node ID according to the application needs. The Node ID can be stored in an electronic medium or changed with the switch.*

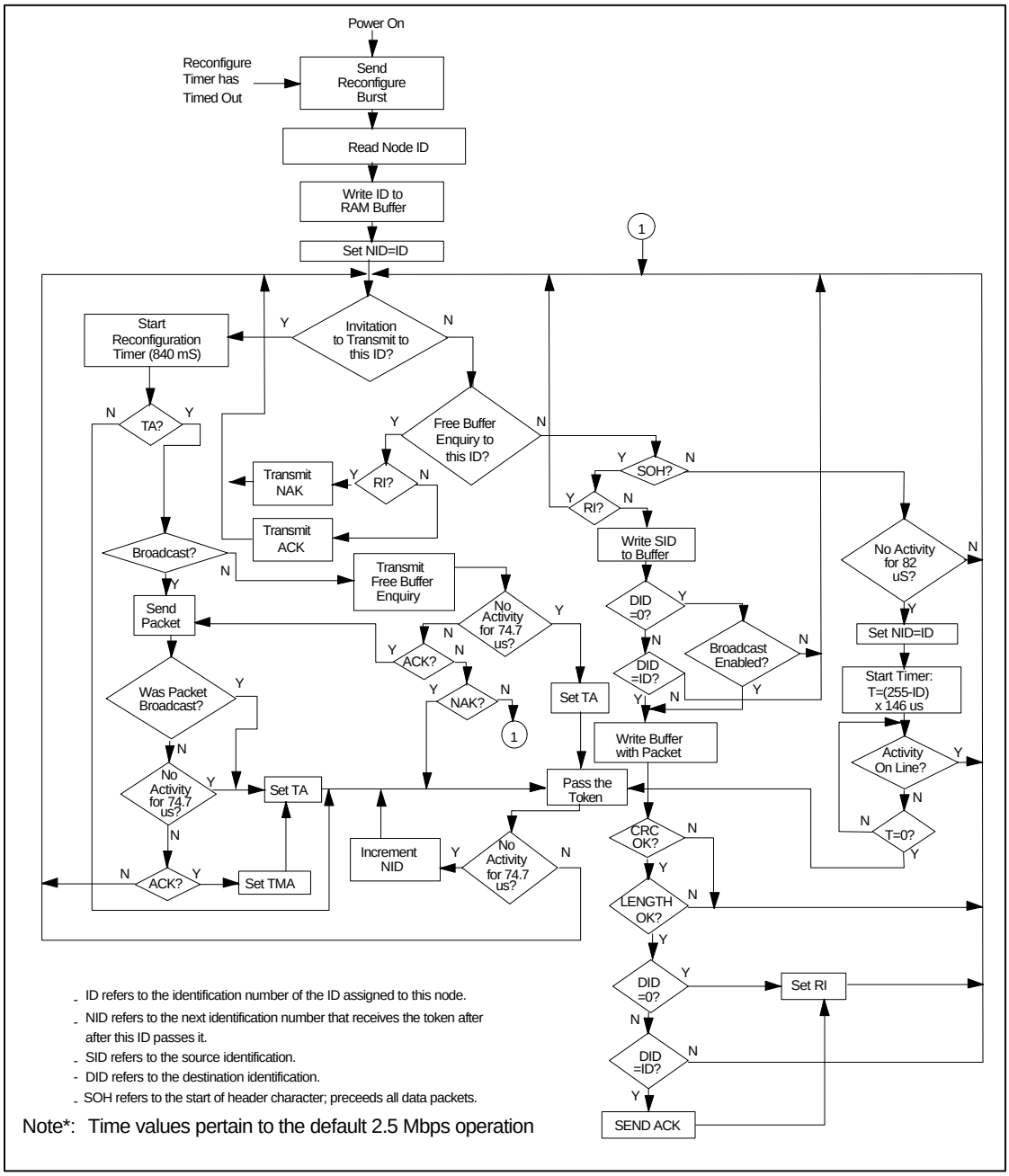


FIGURE 3 - DETAILED ARCNET CORE OPERATION

PROTOCOL DESCRIPTION

NETWORK PROTOCOL

Communication on the network is based on a token passing protocol. Establishment of the network configuration and management of the network protocol are handled entirely by *the COM20051's internal microcoded sequencer called ARCNET network core.* The 80C32 controller core transmits data by simply loading a data packet and its destination ID into the network core's RAM buffer, and issuing a command to enable the transmitter. When the ARCNET core next receives the token, it verifies that the receiving node is ready by first transmitting a FREE BUFFER ENQUIRY message. If the receiving node transmits an ACKnowledge message, the data packet is transmitted followed by a 16-bit CRC. If the receiving node cannot accept the packet (typically its receiver is inhibited), it transmits a Negative AcKnowledge message and the transmitter passes the token. Once it has been established that the receiving node can accept the packet and transmission is complete, the receiving node verifies the packet. If the packet is received successfully, the receiving node transmits an ACKnowledge message (or nothing if it is not received successfully) allowing the transmitter to set the appropriate status bits to indicate successful or unsuccessful delivery of the packet. An interrupt mask permits the ARCNET core to generate an interrupt to the processor when selected status bits become true. Figure 4 is a flow chart illustrating the internal operation of the ARCNET core. *All timing details in the discussion of ARCNET protocol are based on the 2.5 Mbps data rate.*

DATA RATES

The ARCNET core is capable of supporting data rates from 156.25 Kbps to 5 Mbps. For slower or faster data rates, an internal Programmable clock divider scales down the clock frequency. Thus all timeout values are scaled up as shown in the following table:

	DATA RATE		TIMEOUT SCALING FACTOR (MULTIPLY BY)	
CLOCK PRE-SCALER	40 MHz CLOCK DIV. TO 20 MHz	40 MHz UN-DIVIDED	40 MHz CLOCK DIV. TO 20 MHz	40 MHz UN-DIVIDED
÷8	25Mbps	5Mbps	1	.5
÷16	1.25Mbps	25Mbps	2	1
÷32	625Kbps	1.25Mbps	4	2
÷64	3125Kbps	625Kbps	8	4
÷128	156.25 Kbps	3125Kbbps	16	8

NETWORK RECONFIGURATION

A significant advantage of the ARCNET is its ability to adapt to changes on the network. Whenever a new node is activated or deactivated, a NETWORK RECONFIGURATION is performed. When a new ARCNET node is turned on (creating a new active node on the network), or if the COM20051 has not received an INVITATION TO TRANSMIT for 840ms, or if a software reset occurs, the ARCNET node causes a NETWORK 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 network. Since this burst is longer than any other type of transmission, the burst will interfere with the next INVITATION TO TRANSMIT, destroy the token and keep any other node from assuming control of the line.

When any ARCNET node senses an idle line for greater than 82 S, which occurs only when the token is lost, it starts an internal timeout equal to 146 s times the quantity 255 minus its own ID. *The COM20051 starts network reconfiguration by sending an invitation to transmit (TOKEN) first to itself and then to*

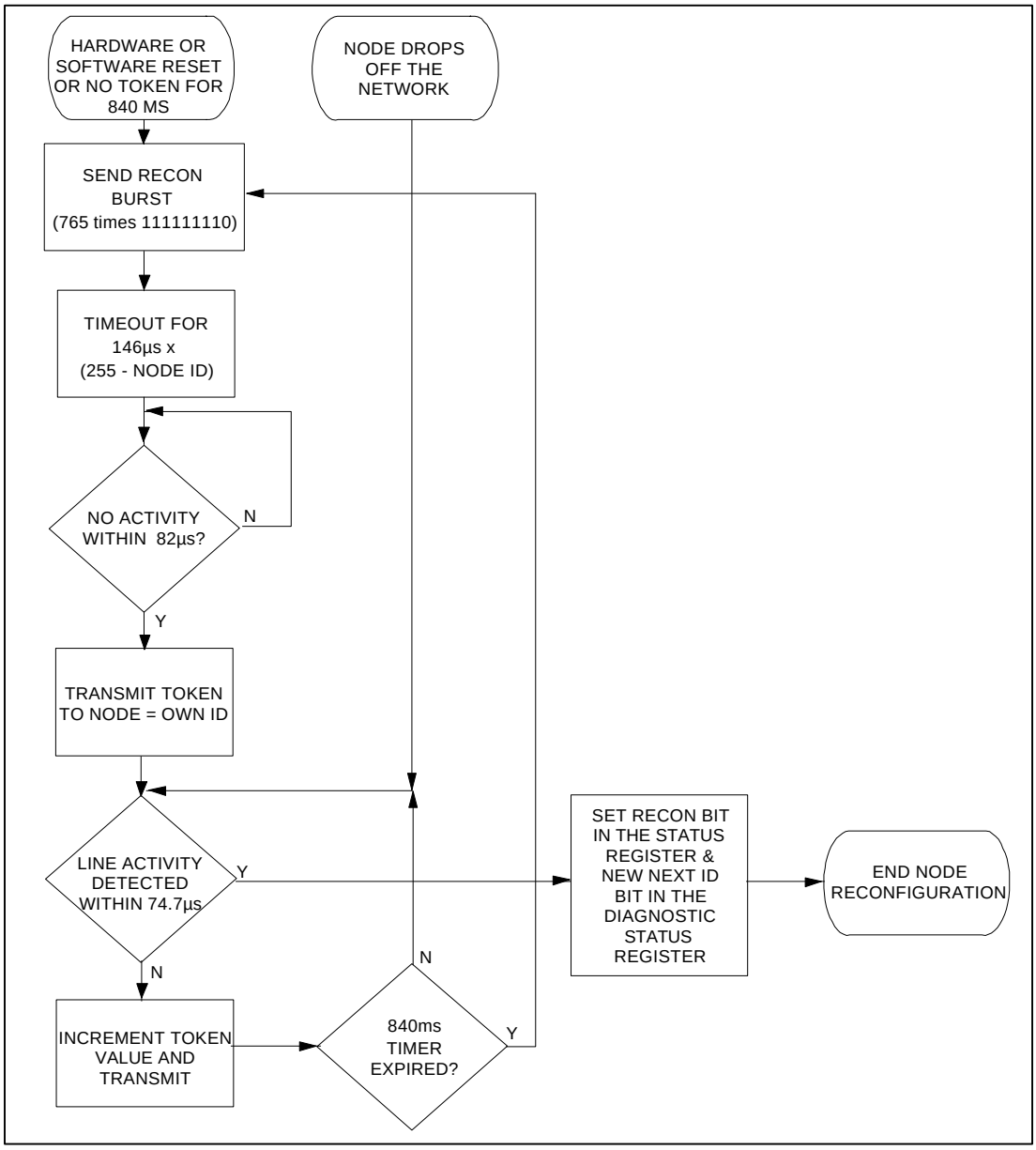


FIGURE 4 - ARCNET RECONFIGURATION PROCESS

all other nodes by incrementing the destination Node ID value. If the timeout expires with no line activity, the ARCNET core starts sending INVITATION TO TRANSMIT with the Destination ID (DID) equal to the currently stored NID. Within a given network, only one node will timeout (the one with the highest ID number). After sending the INVITATION TO TRANSMIT, the COM20051 waits for activity on the line. If there is no activity for 74.7 S, the COM20051 increments the NID value and transmits another INVITATION TO TRANSMIT using the NID equal to the DID. If activity appears before the 74.7 S timeout expires, the COM20051 releases control of the line. During NETWORK RECONFIGURATION, INVITATIONS TO TRANSMIT are sent to all NIDs *(1-255).*

Each COM20051 on the network will finally have saved a NID value equal to the ID of the ARCNET node that it released control to. This is called the Next ID Value. At this point, control is passed directly from one node to the next with no wasted INVITATIONS TO TRANSMIT being sent to ID's not on the network, until the next NETWORK RECONFIGURATION occurs. When a node is powered off, the previous node attempts to pass the token to it by issuing an INVITATION TO TRANSMIT. Since this node does not respond, the previous node times out and transmits another INVITATION TO TRANSMIT to an incremented ID and eventually a response will be received.

The NETWORK RECONFIGURATION time depends on the number of nodes in the network, the propagation delay between nodes, and the highest ID number on the network, but is typically within the range of 24 to 61 ms for 2.5 Mbps operation.

BROADCAST MESSAGES

Broadcasting gives a particular node the ability to transmit a data packet to all nodes on the network simultaneously. *NID=0* is reserved for this feature and no node on the network can be assigned *NID=0.* To broadcast a message, the transmitting node's processor simply loads the RAM buffer with the data packet and sets the DID *(Destination ID)* equal to zero. Figure 12 illustrates the position of each byte in the packet with the DID residing at address *1Hex* of the current page selected in the "Enable Transmit from Page fnn" command. Each individual node has the ability to ignore broadcast messages by setting the most significant bit of the "Enable Receive to Page fnn" command (see Table 8) to a logic "0".

EXTENDED TIMEOUT FUNCTION

There are three timeouts associated with the COM20051 operation. The values of these timeouts are controlled by bits 3 and 4 of the Configuration Register *and bit 5 of the Setup Register (see register description for details).*

Response Time *(ET1, ET2, ET3)*

The Response Time determines the maximum propagation delay allowed between any two nodes, and should be chosen to be larger than the round trip propagation delay between the two furthest nodes on the network plus the maximum turn around time (the time it takes a particular ARCNET node to start sending a message in response to a received message) which is approximately 12.7 S. The round trip propagation delay is a function of the transmission media and network topology. For a typical system using RG62 coax in a baseband system, a one way cable propagation delay of 31 S translates to a distance of about 4 miles. The flow chart in Figure 3 uses a value of 74.7 S (31 + 31 + 12.7) to determine if any node will respond.

Idle Time *(ET1, ET2, ET3)*

The Idle Time is associated with the NETWORK RECONFIGURATION. Figure 3 and Figure 4 illustrate that during a NETWORK RECONFIGURATION one node will continually transmit INVITATIONS TO TRANSMIT until it encounters an active node. All other nodes on the network must distinguish between this operation and an entirely idle line. During NETWORK RECONFIGURATION, activity will appear on the line every 82 S. This 82 S is equal to the Response Time of 74.7 S plus the time it takes the COM20051 to start

retransmitting another message (usually another INVITATION TO TRANSMIT).

Reconfiguration Time *(ET1, ET2)*

If any node does not receive the token within the Reconfiguration Time, it will initiate a NETWORK RECONFIGURATION. The ET2 and ET1 bits of the Configuration Register allow the network to operate over longer distances than the 4 miles stated earlier. The logic levels on these bits control the maximum distances over which the COM20051 can operate by controlling the three timeout values described above. For proper network operation, all nodes connected to the same network must have the same Response Time, Idle Time, and Reconfiguration Time.

LINE PROTOCOL

The ARCNET line protocol is considered isochronous because each byte is preceded by a start interval and ended with a stop interval. Unlike asynchronous protocols, there is a constant amount of time separating each data byte. Each byte takes exactly 11 clock intervals that are defined by nPULSE1 and nPULSE2 signals (at 2.5 Mbps one byte takes 4.4 ms). As a result a time to transmit a message can be precisely determined. The line idles in a spacing (logic "0") condition. A logic "0" is defined as no line activity and a logic "1" is defined as a negative pulse of 200nS duration. A transmission starts with an ALERT BURST consisting of 6 unit intervals of mark (logic "1"). Eight bit data characters are then sent, with each character preceded by 2 unit intervals of mark and one unit interval of space. Five types of transmission can be performed as described below:

Invitations To Transmit *(ITT)*

An Invitation To Transmit is used to pass the token from one node to another and is sent by the following sequence:

- An ALERT BURST
- An *ITT* (*Invitation To Transmit:* ASCII code 04H)
- Two (repeated) DID (Destination ID) characters

ALERT BURST	*ITT*	DID	DID

Free Buffer Enquiries *(FBE)*

A Free Buffer Enquiry is used to ask another node if it is able to accept a packet of data. It is sent by the following sequence:

- An ALERT BURST
- An FBE - Free Buffer Enquiry: ASCII code 85H)
- Two (repeated) DID (Destination ID) characters

ALERT BURST	*FBE*	DID	DID

Data Packets *(PAC)*

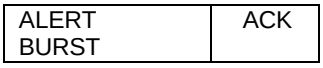
A Data Packet consists of the actual data being sent to another node. It is sent by the following sequence:

- An ALERT BURST
- An *PAC (Data Packet*--ASCII code 01H)
- An SID (Source ID) character
- Two (repeated) DID (Destination ID) characters
- A single COUNT character which is the 2's complement of the number of data bytes to follow if a short packet is sent, or 00*Hex* followed by a COUNT character if a long packet is sent
- N data bytes where COUNT = 256-N (or 512-N for a long packet)
- Two CRC (Cyclic Redundancy Check) characters. The CRC polynomial used is: $X^{16} + X^{15} + X^2 + 1$.

Acknowledgements *(ACK)*

An Acknowledgement is used to acknowledge reception of a packet or as an affirmative response to FREE BUFFER ENQUIRIES and is sent by the following sequence:

- An ALERT BURST
- An ACK (ACKnowledgement--ASCII code 86H) character



Negative Acknowledgements *(NAK)*

A Negative Acknowledgement is used as a negative response to FREE BUFFER ENQUIRIES and is sent by the following sequence:

- An ALERT BURST
- A NAK (Negative Acknowledgement--ASCII code 15H) character

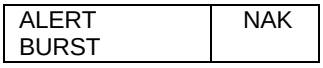


Figure 5 illustrates the flow of events on a 5-node network where a node with the NID=1 transmits a data packet to a node with the NID=5, and a node with the NID=3 tries to transmit a data to a node with the NID=4 but node 4 cannot accept it. All other nodes are just passing the tokens.

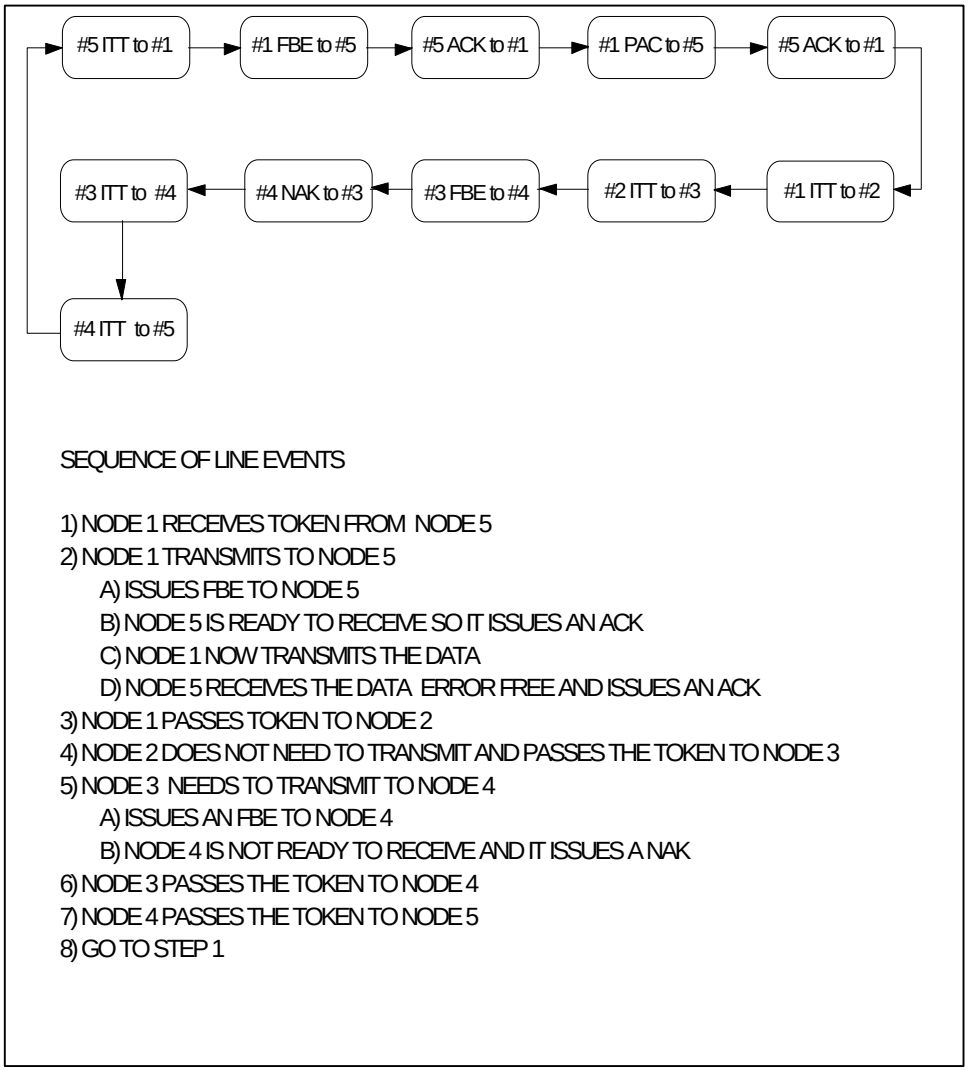


FIGURE 5 – AVERAGE SEQUENCE OF LINE EVENTS FOR A FIVE-NODE NETWORK

SYSTEM DESCRIPTION

MICROCONTROLLER INTERFACE

COM20051 ARCNET network core contains 1K byte of RAM and 11 registers. The internal RAM is accessed via a pointer-based scheme (refer to the Sequential Access Memory section), and the internal registers are accessed via direct addressing. The ARCNET core bus interface is designed to be flexible so that it is independent of the 80C32 speed.

The COM20051 provides for no wait state arbitration via direct addressing to its internal registers and a pointer based addressing scheme to access its internal RAM. *Note that at 5 Mbps data rate, the internal arbiter must be slowed down using the SLOWARB bit of the Setup Register.* The pointer may be used in the auto-increment mode for typical sequential buffer emptying or loading, or it can be taken out of the auto-increment mode to perform out of sequence accesses to the RAM. The data within the RAM is accessed through the Data Register. Data being read is prefetched from memory and placed into the Data Register for the microcontroller to read. During a write operation, the data is stored in the Data Register and then written into memory. Whenever the pointer is loaded for reads with a new value, data is immediately prefetched to prepare for the first read operation.

TRANSMISSION MEDIA INTERFACE

Figure 6 illustrates the COM20051 interface to the transmission media used to connect the node to the network. Table 2 lists different types of cable which are suitable for ARCNET applications.[1] The user may interface to the cable of choice in one of three ways:

1 Please refer to TN7-5 - **Cabling Guidelines for the COM20020 ULANC**, available from SMSC, for recommended cabling distance, termination, and node count for ARCNET nodes.

Traditional Hybrid Interface

The Traditional Hybrid Interface is that which is used with previous ARCNET devices. The Hybrid Interface is recommended if the node is to be placed in a network with other Hybrid-Interfaced nodes. The Traditional Hybrid Interface is for use with nodes operating at 2.5 Mbps only. The transformer coupling of the Hybrid offers isolation for the safety of the system and offers high Common Mode Rejection. The Traditional Hybrid Interface uses circuits like SMSC's HYC9068 or HYC9088 to transfer the pulse-encoded data between the cable and the COM20051. The COM20051 transmits a logic "1" by generating two 100nS non-overlapping negative pulses, nPULSE1 and nPULSE2. Lack of pulses indicates a logic "0". The nPULSE1 and nPULSE2 signals are sent to the Hybrid, which creates a 200nS dipulse signal on the *medium.* During reception, the 200nS dipulse appearing on the media is coupled through the transformer of the LAN Driver, which produces a positive pulse at the RXIN pin of the COM20051. The pulse on the RXIN pin represents a logic "1". Lack of pulse represents a logic "0". Typically, RXIN pulses occur at multiples of 400nS. The COM20051 can tolerate distortion (bit jitter) of plus or minus 100nS and still correctly capture and convert the RXIN pulses to NRZ format. Figure 8 illustrates the events which occur in transmission or reception of data consisting of 1, 1, 0.

Backplane Configuration

The Backplane Configuration is recommended for cost-sensitive, short-distance applications like backplanes and instrumentation. This mode is advantageous because it saves components, cost, and power.

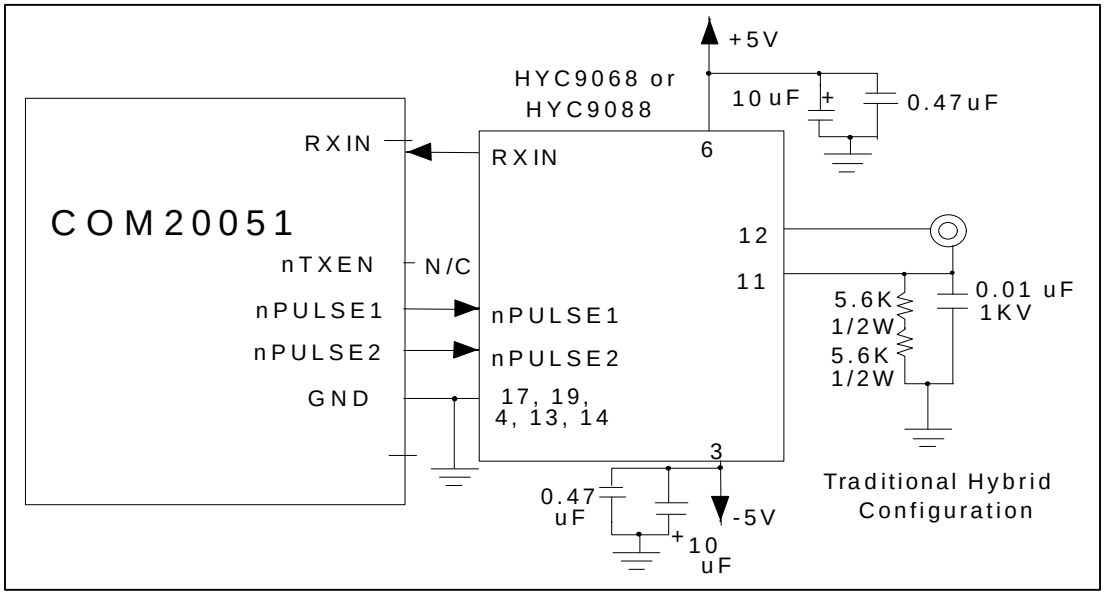


FIGURE 6 – DIPULSE HYBRID CONFIGURATION

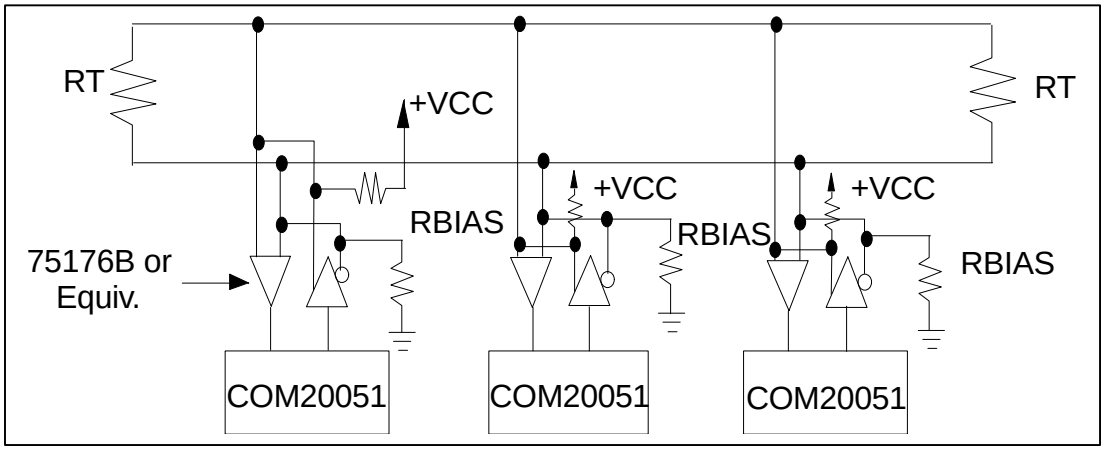


FIGURE 7 – COM20051 NETWORK USING RS-485 DIFFERENTIAL TRANSCEIVERS

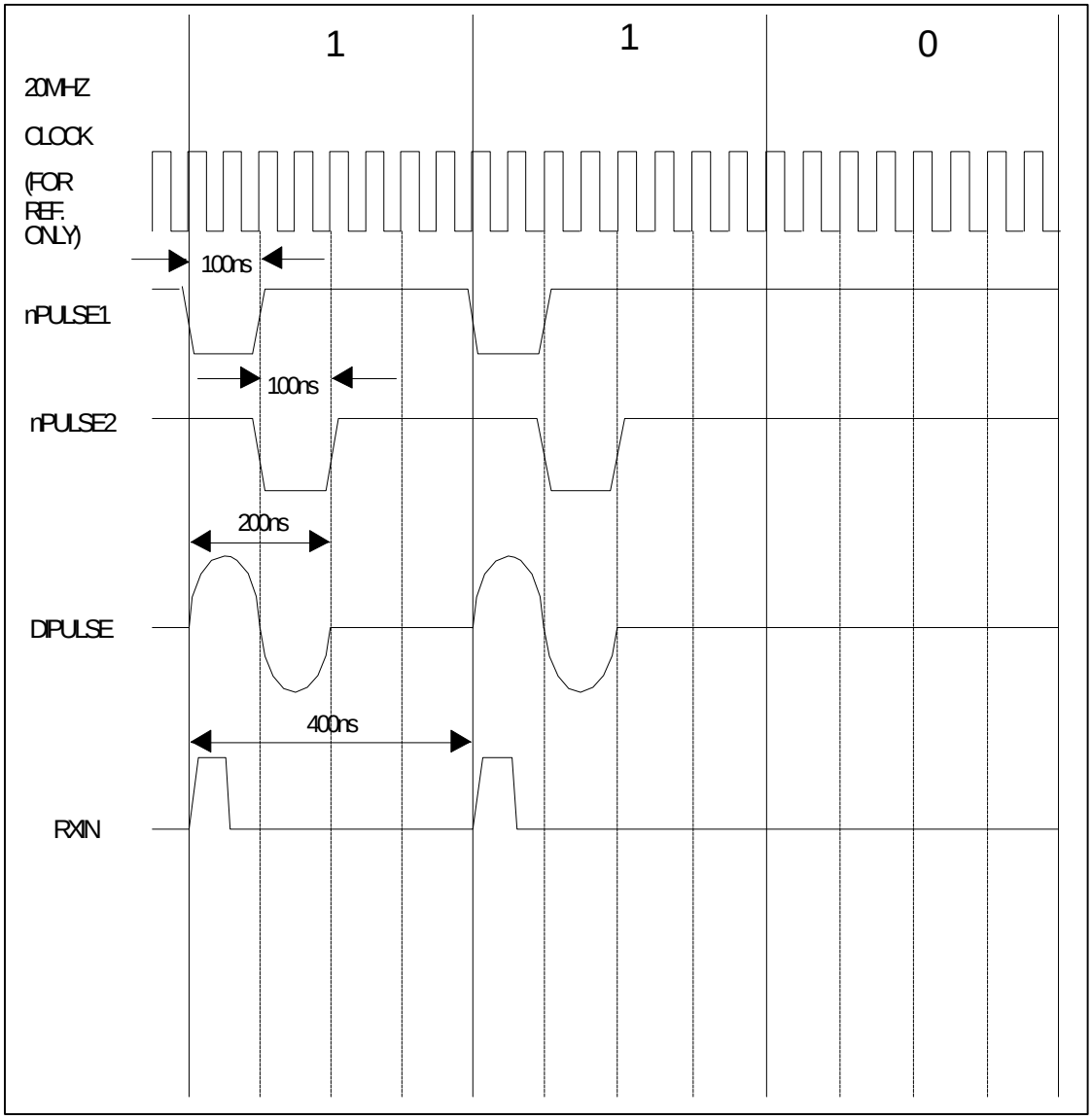


FIGURE 8 – DIPULSE WAVEFORM FOR BIT DATA OF 1-1-0

Since the Backplane Configuration encodes data differently than the traditional Hybrid Configuration, nodes utilizing the Backplane Configuration cannot communicate directly with nodes utilizing the Traditional Hybrid Configuration.

The Backplane Configuration does not isolate the node from the media nor protect it from Common Mode noise, but Common Mode Noise is less of a problem in short distances.

The COM20051 supplies a programmable output driver for Backplane Mode operation. A push/pull or open drain driver can be selected by programming the P1MODE bit of the Setup Register (see register descriptions for details.) The COM20051 defaults to an open drain output.

The Backplane Configuration provides for direct connection between the COM20051 and the *physical medium in open drain configuration of the output driver.* Only one pull-up resistor (for open drain only) is required somewhere on the media (not on each individual node). The nPULSE1 signal, in this mode, is an open drain or push/pull driver and is used to directly drive the media. It issues a 200nS negative pulse to transmit a logic "1". Note that when used in the open-drain mode, the COM20051 does not have a fail/safe input on the RXIN pin.

The nPULSE1 signal actually contains a weak pull-up resistor. This pull-up should not take the place of the resistor required on the media for open drain mode. In typical applications, the serial backplane is terminated at both ends and a bias is provided by the external pull-up resistor.

The RXIN signal is directly connected to the cable via an internal Schmitt trigger. A negative pulse on this input indicates a logic "1". Lack of pulse indicates a logic "0". For typical single-ended backplane applications, RXIN is connected to nPULSE1 to make the serial backplane data line. A ground line (from the coax or twisted pair) should run in parallel with the signal. For applications requiring different treatment of the receive signal (like filtering or squelching), nPULSE1 and RXIN remain as independent pins. External differential drivers/receivers for increased range and common mode noise rejection, for example, would require the signals to be independent of one another. When the device is in Backplane Mode, the clock provided by the nPULSE2 signal may be used for encoding the data into a different encoding scheme or other synchronous operations needed on the serial data stream.

Differential Driver Configuration

The Differential Driver Configuration is a special case of the Backplane Mode. It is a DC coupled configuration recommended for applications like car-area networks (CAN) or other cost-sensitive applications which do not require direct compatibility with existing ARCNET nodes and do not require isolation. *Figure 7 illustrates this configuration.*

The Differential Driver Configuration cannot communicate directly with nodes utilizing the Traditional Hybrid Configuration. Like the Backplane Configuration, the Differential Driver Configuration does not isolate the node from the *physical medium.*

The Differential Driver interface includes a RS485 Driver/Receiver to transfer the data between the cable and the COM20051. The nPULSE1 signal transmits the data, provided the nTXEN signal is active. The nPULSE1 signal issues a 200nS negative pulse to transmit a logic "1". The RXIN pin receives the data. A negative pulse on this input indicates a logic "1". Lack of pulse indicates a logic "0". The transmitter portion of the COM20051 is disabled during reset and the nPULSE1, nPULSE2 and nTXEN pins are inactive.

Table 2 - Typical Media

CABLE TYPE	NOMINAL IMPEDANCE	ATTENUATION PER 1000 FT. AT 5MHZ
RG-62 Belden #86262	93	5.5dB
RG-59/U Belden #89108	75	7.0dB
RG-11/U Belden #89108	75	5.5dB
IBM Type 1* Belden #89688	150	7.0dB
IBM Type 3* Telephone Twisted Pair Belden #1155A	100	17.9dB
COMCODE 26 AWG Twisted Pair Part #105-064-703	105	16.0dB

*Non-plenum-rated cables of this type are also available.

Note: For more detailed information on Cabling options including RS-485, transformer-coupled RS-485 and Fiber Optic interfaces, please refer to TN7-5 - **Cabling Guidelines for the COM20020 ULANC**, available from Standard Microsystems Corporation.

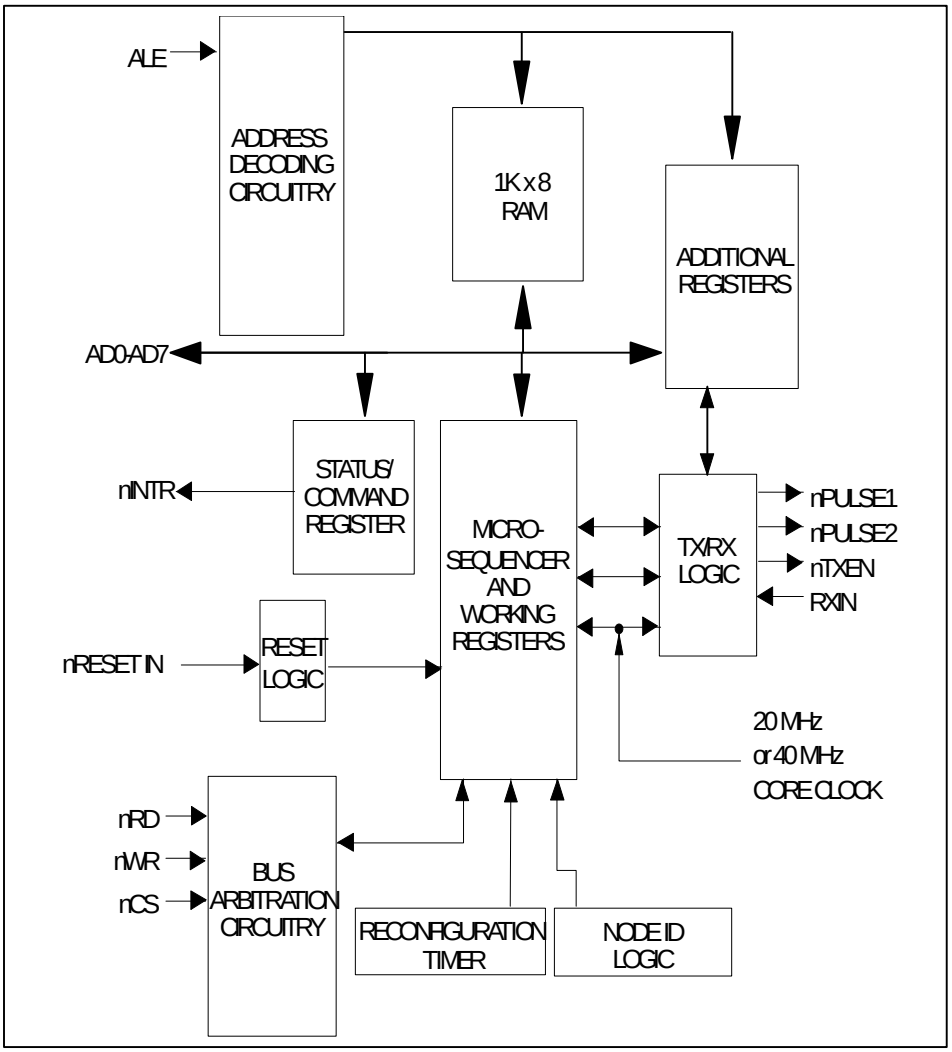


FIGURE 9 – ARCNET CORE BLOCK DIAGRAM

ARCNET CORE FUNCTIONAL DESCRIPTION

MICROSEQUENCER

The ARCNET core contains an internal microsequencer which performs all of the control operations necessary to carry out the ARCNET protocol. It consists of a clock generator, a 544 x 8 ROM, a program counter, two instruction registers, an instruction decoder, a no-op generator, jump logic, and reconfiguration logic.

The ARCNET core derives The Program Counter Clock and Instruction Execution Clock from the SYSTEM CLOCK. If the system clock is 40 MHz the Program Counter Clock runs at 10 MHz and the Instruction Execution Clock runs at 5 MHz. If the System Clock is 20 MHz the above clocks run at 5 MHz and 2.5 MHz respectively. The microprogram is stored in the ROM and the instructions are fetched and then placed into the instruction registers. One register holds the op code, while the other holds the immediate data. Once the instruction is fetched, it is decoded by the internal instruction decoder, at which point the ARCNET core proceeds to execute the instruction. When a no-op instruction is encountered, the microsequencer enters a timed loop and the program counter is temporarily stopped until the loop is complete. When a jump instruction is encountered, the program counter is loaded with the jump address from the ROM. The ARCNET core contains an internal reconfiguration timer which interrupts the microsequencer if it has timed out. At this point the program counter is cleared and the MYRECON bit of the Diagnostic Status Register is set.

INTERNAL REGISTERS

The ARCNET core contains eight internal registers. Tables 4 and 5 illustrate the ARCNET core register map. Reserved locations should not be accessed. All undefined bits are read as undefined and must be written as logic "0".

Interrupt Mask Register (IMR) *(Location +00Hex)*

The ARCNET core is capable of generating an interrupt signal when certain status bits become true. A write to the IMR specifies which status bits will be enabled to generate an interrupt. The bit positions in the IMR are in the same position as their corresponding status bits in the Status Register and Diagnostic Status Register. A logic "1" in a particular position enables the corresponding interrupt. The Status bits capable of generating an interrupt include the Receiver Inhibited bit, New Next ID bit, Excessive NAK bit, Reconfiguration Timer bit, and Transmitter Available bit. No other Status or Diagnostic Status bits can generate an interrupt.

The five maskable status bits are ANDed with their respective mask bits, and the results are ORed to produce the interrupt signal. An RI or TA interrupt is masked when the corresponding mask bit is reset to logic "0", but will reappear when the corresponding mask bit is set to logic "1" again, unless the interrupt status condition has been cleared by this time. A RECON interrupt is cleared when the "Clear Flags" command is issued. An EXCNAK interrupt is cleared when the "POR Clear Flags" command is issued. A New Next ID interrupt is cleared by reading the New Next ID Register. The Interrupt Mask Register defaults to the value 0000 0000 upon hardware reset only. Refer to Table 3 on the following page.

Table 3 - Cleaning Interrupt Bit

Interrupt Type	Cleaning Interrupt Bit
RI	Issue "Enable Receive to Page Fnn" command
EXCNAK	Issue "Clean Flags" Command with "p" bit set
RECON	Issue "Clear Flags" Command with "r" bit set
New Next ID	Read New Next ID Register
TA	Issue "Enable Transmit From Page Fnn" Command

Table 4 – Read Register Summary

REGISTER	READ MSB							LSB	OFFSET ADDRESS
STATUS	RI	X	X	POR	TEST	RECON	TMA	TA	00
DIAG. STATUS	MY-RECON	DUPID	RCVACT	TOKEN	EXCNAK	TENTID	NEW NEXTID	X	01
ADDRESS PTR HIGH	RDDATA	AUTO-INC	X	X	X	X	A9	A8	02
ADDRESS PTR LOW	A7	A6	A5	A4	A3	A2	A1	A0	03
DATA	D7	D6	D5	D4	D3	D2	D1	D0	04
RESERVED	X	X	X	X	X	X	X	X	05
CONFIG-URATION	RESET	CCHEN	TXEN	ET1	ET2	BACK-PLANE	SUB-AD1	SUB-AD0	06
TENTID	TID7	TID6	TID5	TID4	TID3	TID2	TID1	TID0	07
NODEID	NID7	NID6	NID5	NID4	NID3	NID2	NID1	NID0	
SETUP	P1MODE	FOUR NAKS	ET3	RCV_ALL	CKP3	CKP2	CKP1	SLOW ARB	
NEXT ID	NXTID7	NXTID6	NXTID5	NXTID4	NXTID3	NXTID2	NXTID1	NXTID0	
RESERVED	X	X	X	X	X	X	X	X	08
IRR	X	5MBS	DEC3	DEC2	DEC1	EXT	INT1	INT0	09

NOTE: The SLOWARB bit must be set for 5 Mbps operation.

Table 5 – Write Register Summary

OFFSET ADDRESS	WRITE MSB							LSB	REGISTER
00	RI	0	0	0	EXCNAK	RECON	NEW NEXTID	TA	INTERRUPT MASK
01	D7	D6	D5	D4	D3	D2	D1	D0	COMMAND
02	RDDATA	AUTO-INC	0	0	0	0	A9	A8	ADDRESS PTR HIGH
03	A7	A6	A5	A4	A3	A2	A1	A0	ADDRESS PTR LOW
04	D7	D6	D5	D4	D3	D2	D1	D0	DATA
05	0	0	0	0	0	0	0	0	RESERVED
06	RESET	CCHEN	TXEN	ET1	ET2	BACK-PLANE	SUB-AD1	SUB-AD0	CONFIG-URATION
07	TID7	TID6	TID5	TID4	TID3	TID2	TID1	TID0	TENTID
	NID7	NID6	NID5	NID4	NID3	NID2	NID1	NID0	NODEID
	P1MODE	FOUR NAKS	ET3	RCV_ ALL	CKP3	CKP2	CKP1	SLOW ARB	SETUP
	0	0	0	0	0	0	0	0	NEXT ID
08	X	X	X	X	X	X	X	X	RESERVED
09	X	5MBS	DEC3	DEC2	DEC1	EXT	INT1	INT0	IRR

NOTE: The SLOWARB bit must be set for 5 Mbps operation.

Interrupt Routing Register *(Location +09HEX)*

The Interrupt Routing Register (IRR) routes the interrupt generated by the ARCNET core to the appropriate 80C32 interrupt input (INT0 or INT1) or to one of the eight general purpose digital I/O ports (P1.0-1.7) of the 80C32. The interrupt routing operates on a priority driven scheme where if two bits are enabled the highest priority always wins. INT0 has highest priority followed by INT1 then EXT. The nINT0 and nINT1 bits route the interrupt signal to either the nINT0 or nINT1 pin of the 80C32. The 80C32 nINT1 and nINT0 inputs are wire ANDed with the routed interrupt. This allows the 80C32's interrupts to be used for more than one source. If many interrupts are being used in the system, the COM20051 supports the use of an external interrupt controller to arbitrate simultaneous interrupts. External interrupt controllers are supported by programming the EXT bit of the IRR. This will cause the interrupt signal to be present on one of the Port 1 pins as programmed by Bits 3 - 5.

The 5 Mbps bit programs the ARCNET core to operate at a 5 Mbps data rate. The 5 Mbps bit causes the clock to the ARCNET core to double its frequency from 20MHz to 40MHz. 5 Mbps operation requires the SLOWARB bit of the SETUP register to be set. Failure to set the SLOWARB bit may result in errors when accessing the ARCNET buffer RAM.

Table 6 - Interrupt Routing Register

BIT	BIT NAME	SYMBOL	DESCRIPTION
6	5 Mbps Enable	5MBPS	Causes the ARCNET core to operate at a 5 Mbps data rate. Defaults to 0.
3-5	Port 1 Bit Assignment	DEC1 - 3	Selects one of the eight Port 1 bits to output the interrupt on. 000 - P1.0 001 - P1.1 010 - P1.2 011 - P1.3 100 - P1.4 101 - P1.5 110 - P1.6 111 - P1.7 Defaults to 000 (P1.0).
2	External Interrupt Enable	EXT	Enables routing of the ARCNET interrupt onto on the Port 1 pins. Defaults to 0.
1	Interrupt 1 Enable.	INT1	Enables wire Oring of the ARCNET interrupt with the INT1 pin. Defaults to 0.
0	Interrupt 0 Enable.	INT0	Enables wire ORing if the ARCNET interrupt with the INT0 pin. Defaults to 0.

Data Register *(Location +04Hex)*

This read/write 8-bit register is used as the channel through which the data to and from the RAM passes. The data is placed in or retrieved from the address location presently specified by the address pointer. The contents of the Data Register are undefined upon hardware reset. *In the case of READ operation, the Data Register is loaded with the contents of ARCNET Core Internal RAM upon writing Address Pointer Low-only once.*

Tentative ID Register *(Location +07Hex)*

The Tentative ID Register is a read/write 8-bit register accessed when the Sub Address Bits are set up accordingly (please refer to the Configuration Register description). The Tentative ID Register can be used while the node is on-line to build a network map of those nodes existing on the network. It minimizes the need for operator interaction with the network. The node determines the existence of other nodes by placing a Node ID value in the Tentative ID Register and waiting to see if the Tentative ID bit of the Diagnostic Status Register gets set. *The network maps developed by this method has only historical value, since nodes may join or depart from the network at any time.* When using the Tentative ID feature, a node cannot detect the existence of the next logical node to which it passes the token. The Next ID Register will hold the ID value of that node. The Tentative ID Register defaults to the value 0000 0000 upon hardware reset only.

Node ID Register *(Location +07Hex)*

The Node ID Register is a read/write 8-bit register accessed when the Sub Address Bits are set up accordingly (please refer to the Configuration Register). The Node ID Register contains the unique value which identifies this particular node. Each node on the network must occupy a unique Node ID value at all times. The Duplicate ID bit of the Diagnostic Status Register helps the user find a unique Node ID. Refer to the Initialization Sequence section for further detail on the use of the DUPID bit. The microsequencer of the ARCNET core does not wake up until a Node ID other than zero is written into the Node ID Register. During this time, no microcode is executed, no tokens are passed by this node, and no reconfigurations are caused by this node. Once a non-zero Node ID is placed into the Node ID Register, the core wakes up but will not join the network until the TXEN bit of the Configuration Register is set. While the Transmitter is disabled, the Receiver portion of the device is still functional and will provide the user with useful information about the network. The Node ID Register defaults to the value 0000 0000 upon hardware reset only.

Next ID Register *(Location +07Hex)*

The Next ID Register is an 8-bit, read-only register, accessed when the sub-address bits are set up accordingly (please refer to the Configuration Register). The Next ID Register holds the value of the Node ID to which the COM20051 will pass the token. When used in conjunction with the Tentative ID Register, the Next ID Register can provide a complete network map. The Next ID Register is updated each time a node enters/leaves the network or when a network reconfiguration occurs. Each time the microsequencer updates the Next ID Register, a New Next ID interrupt is generated. This bit is cleared by reading the Next ID Register. Default value is 0000 0000 upon hardware or software reset.

Status Register *(Location +00Hex)*

The ARCNET Status Register is an 8-bit read-only register. All of the bits, except for bits 5 and 6, are software compatible with previous SMSC ARCNET devices. In previous SMSC ARCNET devices the Extended Timeout status was provided in bits 5 and 6 of the Status Register. In the COM20020, the COM20020-5, COM20010, COM90C66, and the COM90C165, these bits exist in and are controlled by the Configuration Register. The Status Register contents are defined as in Table 6, but are defined differently during the Command Chaining operation. Please refer to the Command Chaining section for the definition of

the Status Register during Command Chaining operation. The Status Register defaults to the value 1XX1 0001 upon either hardware or software reset.

Diagnostic Status Register *(Location +01Hex)*

The Diagnostic Status Register contains seven read-only bits which help the user troubleshoot the network or node operation. Various combinations of these bits and the TXEN bit of the Configuration Register represent different situations. All of these bits, except the Excessive NAK bit and the New Next ID bit, are reset to logic "0" upon reading the Diagnostic Status Register or upon software or hardware reset. The EXCNAK bit is reset by the "POR Clear Flags" command upon software or hardware reset. The Diagnostic Status Register defaults to the value 0000 000X upon either hardware or software reset.

Command Register *(Location +01Hex)*

Execution of commands are initiated by performing *write operations* to this register. Any combinations of written data other than those listed in Table 8 are not permitted and may result in incorrect chip and/or network operation.

Address Pointer Registers *(Location +02Hex, +03Hex)*

These read/write registers are each 8-bits wide and are used for addressing the internal ARCNET RAM. New pointer addresses should be written by first writing to the High Register and then writing to the Low Register because writing to the Low Register loads the address. The contents of the Address Pointer High and Low Registers are undefined upon hardware reset.

Configuration Register *(Location +06Hex)*

The Configuration Register is a read/write register which is used to configure the different modes of the ARCNET core. The Configuration Register defaults to the value 0001 1000 upon hardware reset only.

Setup Register *(Location +07Hex)*

The Setup Register is a read/write 8-bit register accessed when the Sub Address Bits are set up accordingly (see the bit definitions of the Configuration Register). The Setup Register allows the user to change the network speed (data rate) or the arbitration speed independently, invoke the Receive All feature, change the nPULSE1 driver type, and reduce protocol timeouts by a factor of 3. The data rate may be slowed to 156.25 Kbps and/or the arbitration speed may be slowed by a factor of two. *The SLOWARB must be set to a 1 for the 5 Mbps operation.* The Setup Register defaults to the value 0000 0000 upon hardware reset only.

Table 7 - Status Register

BIT	BIT NAME	SYMBOL	DESCRIPTION
7	Receiver Inhibited	RI	This bit, if high, indicates that the receiver is not enabled because either an "Enable Receive to Page fnn" command was never issued, or a packet has been deposited into the RAM buffer page fnn as specified by the last "Enable Receive to Page fnn" command. No messages will be received until this command is issued, and once the message has been received, the RI bit is set, thereby inhibiting the receiver. The RI bit is cleared by issuing an "Enable Receive to Page fnn" command. This bit, when set, will cause an interrupt if the corresponding bit of the Interrupt Mask Register (IMR) is also set. When this bit is set and another station attempts to send a packet to this station, this station will send a NAK.
6,5	(Reserved)		These bits are undefined.
4	Power On Reset	POR	This bit, if high, indicates that the ARCNET core has been reset by either a software reset, a hardware reset, or writing 00H to the Node ID Register. The POR bit is cleared by the "Clear Flags" command.
3	Test	TEST	This bit is intended for test and diagnostic purposes. It is a logic "0" under normal operating conditions.
2	Reconfiguration	RECON	This bit, if high, indicates that the Line Idle Timer has timed out because the RXIN pin was idle for 82 S. The RECON bit is cleared during a "Clear Flags" command. This bit, when set, will cause an interrupt if the corresponding bit in the IMR is also set. The interrupt service routine should consist of examing the MYRECON bit of the Diagnostic Status Register to determine whether there are consecutive reconfigurations caused by this node.
1	Transmitter Message Acknowledged	TMA	This bit, if high, indicates that the packet transmitted as a result of an "Enable Transmit from Page fnn" command has been acknowledged. This bit should only be considered valid after the TA bit (bit 0) is set. Broadcast messages are never acknowledged. The TMA bit is cleared by issuing the "Enable Transmit from Page fnn" command.
0	Transmitter Available	TA	This bit, if high, indicates that the transmitter is available for transmitting. This bit is set when the last byte of scheduled packet has been transmitted out, or upon execution of a "Disable Transmitter" command. The TA bit is cleared by issuing the "Enable Transmit from Page fnn" command after the node next receives the token. This bit, when set, will cause an interrupt if the corresponding bit in the IMR is also set.

Table 8 - Diagnostic Status Register

BIT	BIT NAME	SYMBOL	DESCRIPTION
7	My Reconfiguration	MY-RECON	This bit, if high, indicates that a past reconfiguration was caused by this node. It is set when the Lost Token Timer times out, and is typically read following an interrupt caused by RECON. Refer to the Improved Diagnostics section for further detail.
6	Duplicate ID	DUPID	This bit, if high, indicates that the value in the Node ID Register matches both Destination ID characters of the token and a response to this token has occurred. The EOT character and the trailing zero's are also verified. A logic "1" on this bit indicates a duplicate Node ID, thus the user should write a new value into the Node ID Register. **This bit is only useful for duplicate ID detection when the device is off line, that is, when the transmitter is off.** When the device is on line it will be set every time the device gets the token. This bit is reset automatically upon reading the Diagnostic Status Register. Refer to the Improved Diagnostics section for further detail.
5	Receive Activity	RCVACT	This bit, if high, indicates that data activity (logic "1") was detected on the RXIN pin of the device. Refer to the Improved Diagnostics section for further detail.
4	Token Seen	TOKEN	This bit, if high, indicates that a token has been seen on the network, sent by a node other than this one. Refer to the Improved Diagnostic section for further detail.
3	Excessive NAK	EXCNAK	This bit, if high, indicates that either 128 or 4 Negative Acknowledgements have occurred in response to the Free Buffer Enquiry. This bit is cleared upon the "POR Clear Flags" command. Reading the Diagnostic Status Register does not clear this bit. This bit, when set, will cause an interrupt if the corresponding bit in the IMR is also set. Refer to the Improved Diagnostics section for further detail.
2	Tentative ID	TENTID	This bit, if high, indicates that a response to a token whose DID matches the value in the Tentative ID Register has occurred. In addition, the EOT character is checked. The second DID and the trailing zero's are not checked. Since each node sees every token passed around the network, this feature can be used with the device on-line in order to build and update a network map. Refer to the Improved Diagnostics section for further detail.
1	New Next ID	NEW NXTID	This bit, if high, indicates that the Next ID Register has been updated and that a node has either joined or left the network. Reading the Diagnostic Status Register does not clear this bit. This bit, when set, will cause an interrupt if the corresponding bit in the IMR is also set. The bit is cleared by reading the Next ID Register.
1,0	(Reserved)		These bits are undefined.

Table 9 - Command Register

DATA	COMMAND	DESCRIPTION
0000 0000	Clear Transmit Interrupt	This command is used only in the Command Chaining operation. Please refer to the Command Chaining section for definition of this command.
0000 0001	Disable Transmitter	This command will cancel any pending transmit command (transmission that has not yet started) and will set the TA (Transmitter Available) status bit to logic "1" when the ARCNET core next receives the token.
0000 0010	Disable Receiver	This command will cancel any pending receive command. If the COM20051 is not yet receiving a packet, the RI (Receiver Inhibited) bit will be set to logic "1" the next time the token is received. If packet reception is already underway, reception will run to its normal conclusion.
b0fn n100	Enable Receive to Page fnn	This command allows the ARCNET core to receive data packets into RAM buffer page fnn and resets the RI status bit to logic "0". The values placed in the "nn" bits indicate the page that the data will be received into (page 00 or 01). If the value of "f" is a logic "1", an offset of 256 bytes will be added to that page specified in "nn", allowing a finer resolution of the buffer. Refer to the Selecting RAM Page Size section for further detail. If the value of "b" is logic "1", the device will also receive broadcasts (transmissions to ID zero). The RI status bit is set to logic "1" upon successful reception of a message.
00fn n011	Enable Transmit from Page fnn	This command prepares the ARCNET core to begin a transmit sequence from RAM buffer page fnn the next time it receives the token. The values of the "nn" bits indicate which page to transmit from (0 or 1). If "f" is logic "1", an offset of 256 bytes is added to that page specified in "nn", allowing a finer resolution of the buffer. Refer to the Selecting RAM Page Size section for further detail. When this command is loaded, the TA and TMA bits are reset to logic "0". The TA bit is set to logic "1" upon completion of the transmit sequence. The TMA bit will have been set by this time if the device has received an ACK from the destination node. The ACK is strictly hardware level, sent by the receiving node before its microcontroller is even aware of message reception. Refer to Figure 3 for details of the transmit sequence and its relation to the TA and TMA status bits.
0000 c101	Define Configuration	This command defines the maximum length of packets that may be handled by the device. If "c" is a logic "1", the device handles both long and short packets. If "c" is a logic "0", the device handles only short packets.
000r p110	Clear Flags	This command resets certain status bits of the COM20051. A logic "1" on "p" resets the POR status bit and the EXCNAK Diagnostic status bit. A logic "1" on "r" resets the RECON status bit.
0000 1000	Clear Receive Interrupt	This command is used only in the Command Chaining operation. Please refer to the Command Chaining section for definition of this command.

Table 10 - Address Pointer High Register

BIT	BIT NAME	SYMBOL	DESCRIPTION
7	Read Data	RDDATA	This bit tells the ARCNET core whether the following access will be a read or write. A logic "1" prepares the device for a read, a logic "0" prepares it for a write.
6	Auto Increment	AUTOINC	This bit controls whether the address pointer will increment automatically. A logic "1" on this bit allows automatic increment of the pointer after each access, while a logic "0" disables this function. Please refer to the Sequential Access Memory section for further detail.
5-2	(reserved)		These bits are undefined.
1-0	Address 9-8	A9-A8	These bits hold the upper two address bits which provide addresses to RAM.

Table 11 - Address Pointer Low Register

BIT	BIT NAME	SYMBOL	DESCRIPTION
7-0	Address 7-0	A7-A0	These bits hold the lower 8 address bits which provide the addresses to RAM.

Table 12 - Configuration Register

BIT	BIT NAME	SYMBOL	DESCRIPTION
7	Reset	RESET	A software reset of the ARCNET core is executed by writing a logic "1" to this bit. *A software reset does not reset the microcontroller interface mode, nor does it affect the Configuration Register.* The only registers that the software reset affect are the Status Register, the Next ID Register, and the Diagnostic Status Register. This bit must be brought back to logic "0" to release the reset. *The Software Reset applies only to the ARCNET core. It does not apply to the 8051 microcontroller of the COM20051.*
6	Command Chaining Enable	CCHEN	This bit, if high, enables the Command Chaining operation of the device. Please refer to the Command Chaining section for further details. A low level on this bit ensures software compatibility with previous SMSC ARCNET devices.
5	Transmit Enable	TXEN	When low, this bit disables transmissions by keeping nPULSE1, nPULSE2 if in non-Backplane Mode, and nTXENABLE inactive. When high, it enables the above signals to be activated during transmissions. This bit defaults low upon reset. This bit is typically enabled once the Node ID is determined, and never disabled during normal operation. Please refer to the Improved Diagnostics section for details on evaluating network activity.

Table 12 - Configuration Register

<table>
<tr><th>BIT</th><th>BIT NAME</th><th>SYMBOL</th><th>DESCRIPTION</th></tr>
<tr><td>4,3</td><td>Extended Timeout 1,2</td><td>ET1, ET2</td><td>These bits allow the network to operate over longer distances than the default maximum 4 miles by controlling the Response, Idle, and Reconfiguration Times. All nodes should be configured with the same timeout values for proper network operation. The bit combinations follow.

2.5 Mbps operation
<table>
<tr><th>ET2</th><th>ET1</th><th>Response Time (µS)</th><th>Idle Time (µS)</th><th>Reconfig Time (mS)</th></tr>
<tr><td>0</td><td>0</td><td>1193.6</td><td>1312</td><td>1680</td></tr>
<tr><td>0</td><td>1</td><td>596.8</td><td>656</td><td>1680</td></tr>
<tr><td>1</td><td>0</td><td>298.4</td><td>328</td><td>1680</td></tr>
<tr><td>1</td><td>1</td><td>74.7</td><td>82</td><td>840</td></tr>
</table>
5 Mbps operation
<table>
<tr><th>ET2</th><th>ET1</th><th>Response Time (µS)</th><th>Idle Time (µS)</th><th>Reconfig Time (mS)</th></tr>
<tr><td>0</td><td>0</td><td>596.8</td><td>656</td><td>840</td></tr>
<tr><td>0</td><td>1</td><td>298.4</td><td>328</td><td>840</td></tr>
<tr><td>1</td><td>0</td><td>74.7</td><td>82</td><td>840</td></tr>
<tr><td>1</td><td>1</td><td>37.35</td><td>41</td><td>420</td></tr>
</table></td></tr>
<tr><td>2</td><td>Backplane</td><td>BACK-PLANE</td><td>A logic "1" on this bit puts the device into Backplane Mode signalling which is used for Open Drain and Differential Driver interfaces.</td></tr>
<tr><td>1,0</td><td>Sub Address 1,0</td><td>SUBAD 1,0</td><td>These bits determine which register at address 07 may be accessed. The combinations are as follows:

<table>
<tr><th>SUBAD1</th><th>SUBAD0</th><th>Register</th></tr>
<tr><td>0</td><td>0</td><td>Tentative ID</td></tr>
<tr><td>0</td><td>1</td><td>Node ID</td></tr>
<tr><td>1</td><td>0</td><td>Setup</td></tr>
<tr><td>1</td><td>1</td><td>Next ID</td></tr>
</table></td></tr>
</table>

Table 13 - Setup Register

BIT	BIT NAME	SYMBOL	DESCRIPTION
7	Pulse1 Mode	P1MODE	This bit determines the type of PULSE1 output driver used in Backplane Mode. When high, a push/pull output is used. When low, an open drain output is used. The default is open drain.
6	Four NACKS	FOUR NACKS	This bit, when set, will cause the EXNACK bit in the Diagnostic Status Register to set after four NACKs to Free Buffer Enquiry are detected by the ARCNET core. This bit, when reset, will set the EXNACK bit after 128 NACKs to Free Buffer Enquiry. The default is 128.
5	ET3	ET3	This bit, when set, scales down protocol timeout values *of Response Time and Idle Time but not Reconfiguration Time* to optimize network performance in short topologies. Provides a scaling factor of ÷ 3. Defaults to a zero. Must be reset to be ARCNET compliant.
4	Receive All	RCVALL	This bit, when set, allows the COM20051 to receive all valid data packets on the network, regardless of their destination ID. This mode can be used to implement a network monitor with the transmitter on- or off-line. Note that ACKs are only sent for packets received with a destination ID equal to the COM20051's programmed node ID. This feature can be used to put the COM20051 in a 'listen-only' mode, where the transmitter is disabled and the COM20051 is not passing tokens. Defaults low.
3,2,1	Clock Prescaler Bits *3,2,1*	CKP2,1,0	These bits are used to determine the data rate of the COM20051. The following table is for a 40MHz crystal: Response CKP3 CKP2 CKP1 DIVISOR CLOCK 0 0 0 8 2.5 Mbps 0 0 1 16 1.25 Mbps 0 1 0 32 625 Kbps 0 1 1 64 312.5 Kbps 1 0 0 128 156.25 Kbps 1 0 1 256 Reserved 1 1 0 Reserved 1 1 1 Reserved NOTE: The lowest data rate achievable by the COM20051 is 156.25 Kbps. A divide by 256 is provided for those systems that use faster clock speeds. Defaults to 000 or 2.5 Mbps.
0	Slow Arbitration Select	SLW-ARB	This bit, when set, will divide the arbitration clock by 2. Memory cycle times will increase when slow arbitration is selected. **NOTE: For 5 Mbps operation, SLOWARB must be set.** Defaults to low.

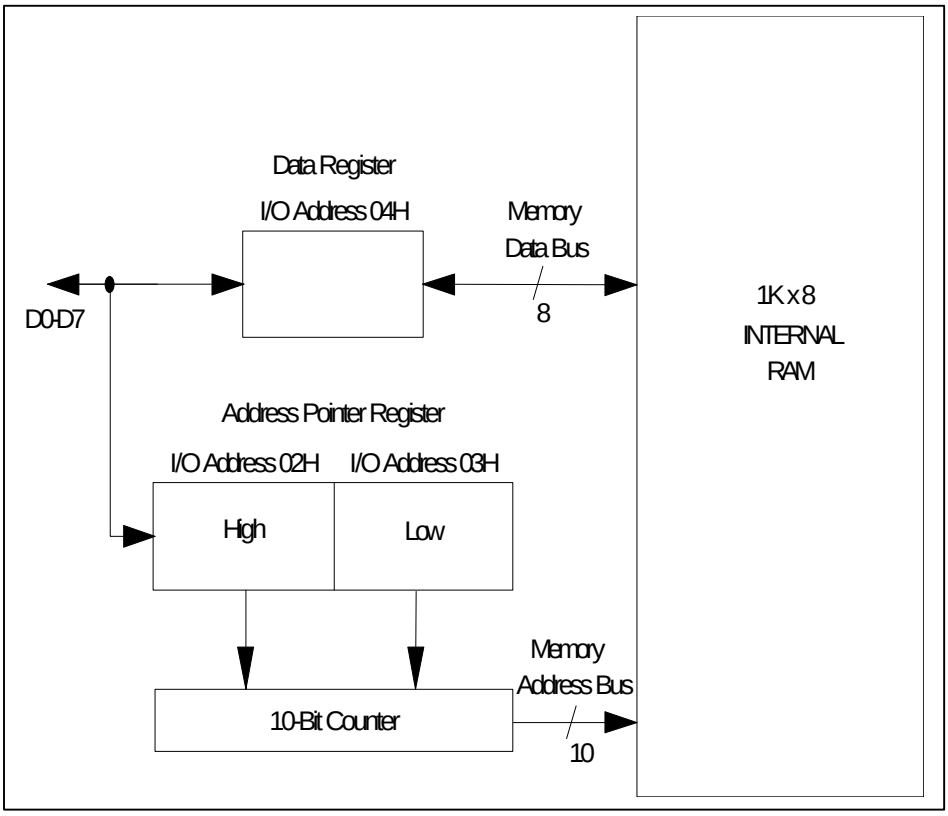


FIGURE 10 – SEQUENTIAL ACCESS OPERATION

INTERNAL RAM

The integration of the 1K x 8 RAM in the ARCNET core represents significant real estate savings. The PC board is now free of the cumbersome external RAM, external latch, and multiplexed address/data bus and control functions which were necessary to interface to the RAM.

The integration of RAM represents significant cost savings because it isolates the system designer from the changing costs of external RAM and it minimizes reliability problems, assembly time and costs, and layout complexity.

Sequential Access Memory

The internal RAM is accessed via a pointer-based scheme. Rather than interfering with system memory, the internal RAM is indirectly accessed through the Address High and Low Pointer Registers. The data is channeled to and from the microcontroller via the 8-bit Data Register. For example: a packet in the internal RAM buffer is read by the microcontroller by writing the corresponding address into the Address Pointer High and Low Registers (offsets 02H and 03H). Note that the High Register should be written first, followed by the Low Register, because writing to the Low Register loads the address. At this point the device accesses that location and places the corresponding data into the Data Register. The microcontroller then reads the Data Register (offset 04H) to obtain the data at the specified location. If the Auto Increment bit is set to logic "1", the device will automatically increment the address and place the next byte of data into the Data Register, again to be read by the microcontroller. This process is continued until the entire packet is read out of RAM. Refer to Figure 10 for an illustration of the Sequential Access operation.

When switching between reads and writes, the pointer must first be written with the starting address. At least one cycle time should separate the pointer being loaded and the first read (see timing parameters).

Access Speed

The ARCNET core is able to accommodate very fast access cycles to its registers and buffers. Arbitration to the buffer does not slow down the cycle because the pointer based access method allows data to be prefetched from memory and stored in a temporary register. Likewise, data to be written is stored in the temporary register and then written to memory.

A Slow Arbitration Bit is provided in the Setup Register to slow down the arbitration clock for buffer accesses at 5 Mbps. The SLOWARB bit must be set to a "1" for 5 Mbps operation.

SOFTWARE INTERFACE

The 80C32 core interfaces to the ARCNET core via software by accessing the various registers. These actions are described in the Internal Registers section. The software flow for accessing the data buffer is based on the Sequential Access scheme. The basic sequence is as follows:

- Disable Interrupts
- Write to Pointer Register High (specifying Auto-Increment mode.)
- Write to Pointer Register Low (this loads the address.)
- Enable Interrupts
- Read or write the Data Register (repeat as many times as necessary to empty or fill the buffer.)

- The pointer may now be read to determine how many transfers were completed.

The software flow for controlling the Configuration, Node ID, Tentative ID, and Next ID registers is generally limited to the initialization sequence and the maintenance of the network map.

Additionally, it is necessary to understand the details of how the other Internal Registers are used in the transmit and receive sequences and to know how the internal RAM buffer is properly set up. The sequence of events that tie these actions together is discussed as follows.

Selecting RAM Page Size

During normal operation, the 1K x 8 of RAM is divided into two pages of 512 bytes each. The page to be used is specified in the "Enable Transmit (Receive) from (to) Page fnn" command, where "nn" specifies page 0 or 1. This allows the user to have constant control over the allocation of RAM.

When the Offset bit "f" (bit 5 of the "Enable Transmit (Receive) from (to) Page fnn" command word) is set to logic "1", an offset of 256 bytes is added to the page specified. For example: to transmit from the second half of page 0, the command "Enable Transmit from Page fnn" (fnn=100 in this case) is issued by writing 0010 0011 to the Command Register. This allows a finer resolution of the buffer pages without affecting software compatibility. This scheme is useful for applications which frequently use packet sizes of 256 bytes or less, especially for microcontroller systems with limited memory capacity. The remaining portions of the buffer pages which are not allocated for current transmit or receive packets may be used as temporary storage for previous network data, packets to be sent later, or as extra memory for the system, which may be indirectly accessed.

If the device is configured to handle both long and short packets (see "Define Configuration" command), then the receive page should always be 512 bytes long because the user never knows what the length of the receive packet will be. In this case, the transmit page may be made 256 bytes long, leaving at least 256 bytes free at any given time. Please note that it is the responsibility of software to reserve 512 bytes for the receive page if the device is configured to handle long packets. The ARCNET core does not check page boundaries during reception.

If the device is configured to handle only short packets, then both transmit and receive pages may be allocated as 256 bytes long, allowing two receive and two transmit packets.

The general rule which may be applied to determine where in RAM a page begins is as follows:

Address = (nn x 512) + (f x 256).

Transmit Sequence

During a transmit sequence, the microcontroller selects a 256 or 512 byte segment of the RAM buffer and writes into it. The appropriate buffer size is specified in the "Define Configuration" command. When long packets are enabled, the ARCNET core interprets the packet as either a long or short packet, depending on whether the buffer address 2 contains a zero or non-zero value. The format of the buffer is shown in Figure 11. Address 0 contains the Source Identifier (SID); Address 1 contains the Destination Identifier (DID); Address 2 (COUNT) contains, for short packets, the value 256-N, where *N represents the number of information bytes in the message,* or for long packets, the value 0, indicating that it is indeed a long packet. In the latter case, Address 3 (COUNT) would contain the value 512-N, where *N represents the number of information bytes in the message.* The SID in Address 0 is used by

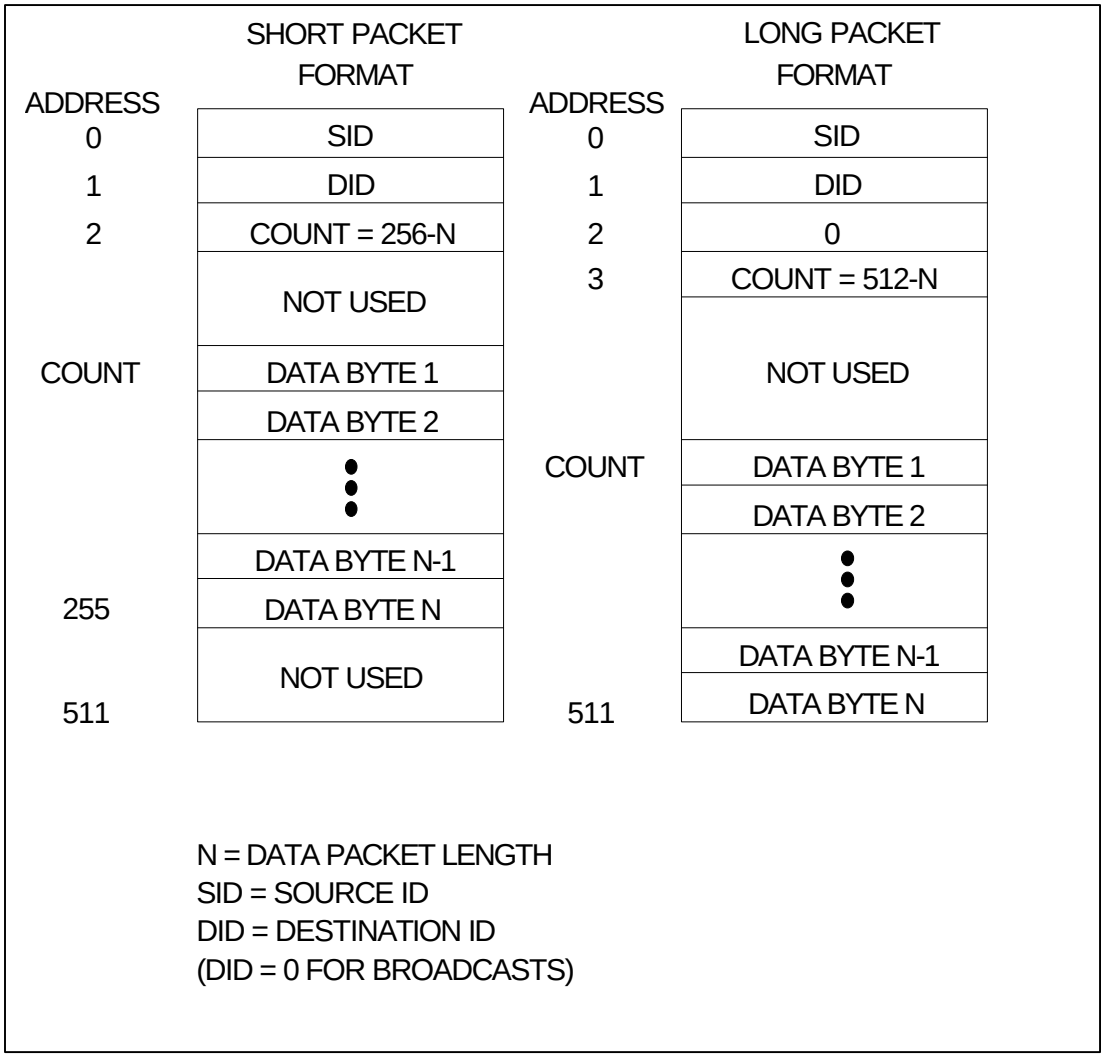


FIGURE 11 – RAM BUFFER PACKET CONFIGURATION

the receiving node to reply to the transmitting node. The ARCNET core puts the local ID in this location, therefore it is not necessary to write into this location.

Please note that a short packet may contain between 1 and 253 data bytes, while a long packet may contain between 257 and 508 data bytes. A minimum value of 257 exists on a long packet so that the COUNT is expressable in eight bits. This leaves three exception packet lengths which do not fit into either a short or long packet; packet lengths of 254, 255, or 256 bytes. If packets of these lengths must be sent, the user must add dummy bytes to the packet in order to make the packet fit into a long packet. Note that only the number of bytes specified in the byte count plus the three-byte header are transmitted. For example, if the byte count is equal to 253, only three bytes of data will be transmitted plus the header (SID, DID, Byte Count) for a total of six bytes.

Once the *packet is written into the buffer,* the microcontroller awaits a logic "1" on the TA bit, indicating that a previous transmit command has concluded and another may be issued. Each time the message is loaded and a transmit command issued, it will take a variable amount of time before the message is transmitted, depending on the traffic on the network and the location of the token at the time the transmit command was issued. *The conclusion of the Transmit command will generate an interrupt if the Interrupt Mask allows it.* If the device is configured for the Command Chaining operation, please see the Command Chaining section for further detail on the transmit sequence. Once the TA bit becomes a logic "1", the microcontroller *may issue* the "Enable Transmit from Page fnn" command, which resets the TA and TMA bits to logic "0". If the message is not a BROADCAST, the ARCNET core automatically *sends* a FREE BUFFER ENQUIRY to the destination node in order to send the message. At this point, one of four possibilities may occur.
The first possibility is if a free buffer is available at the destination node, in which case it responds with an ACKnowledgement. At this point, the ARCNET core fetches the data from the Transmit Buffer and performs the transmit sequence. If a successful transmit sequence is completed, the TMA bit and the TA bit are set to logic "1". If the packet was not transmitted successfully, TMA will not be set. A successful transmission occurs when the receiving node responds to the packet with an ACK. An unsuccessful transmission occurs when the receiving node does not respond to the packet.

The second possibility is if the destination node responds to the Free Buffer Enquiry with a Negative AcKnowledgement. A NAK occurs when the RI bit of the destination node is a logic "1". In this case, the token is passed on from the transmitting node to the next node. The next time the transmitter receives the token, it will again transmit a FREE BUFFER ENQUIRY. If a NAK is again received, the token is again passed onto the next node. The Excessive NAK bit of the Diagnostic Status Register is used to prevent an endless *sending* of FBE's and NAK's. If no *limit of FBE-NAK sequences existed,* the transmitting node would continue issuing a Free Buffer Enquiry, even though it would continuously receive a NAK as a response. The EXCNAK bit generates an interrupt (if enabled) in order to tell the microcontroller to disable the transmitter via the "Disable Transmitter" command. This causes the transmission to be abandoned and the TA bit to be set to a logic "1" when the node next receives the token, while the TMA bit remains at a logic "0". Please refer to the Improved Diagnostics section for further detail on the EXCNAK bit.

The third possibility which may occur after a FREE BUFFER ENQUIRY is issued is if the destination node does not respond at all. In this case, the TA bit is set to a logic "1", while the TMA bit remains at a logic "0". The user should determine whether the node should try to reissue the transmit command.
The fourth possibility is if a non-traditional response is received (some pattern other than ACK or NAK, such as noise). In this case, the token is not passed onto the next node, which causes the Lost Token Timer of the next node to time out, thus generating a network reconfiguration.

The "Disable Transmitter" command may be used to cancel any pending transmit command when the ARCNET core next receives the token. Normally, in an active network, this command will set the TA status bit to a logic "1" when the token is received. If the "Disable Transmitter" command does not cause the TA bit to be set in the time it takes the token to make a round trip through the network, one of three situations exists. Either the node is disconnected from the network, or there are no other nodes on the network, or the external receive circuitry has failed. These situations can be determined by either using the improved diagnostic features of the ARCNET core or using another software timeout which is greater than the worst case time for a round trip token pass, which occurs when all nodes transmit a maximum length message.

Receive Sequence

A receive sequence begins with the RI status bit becoming a logic "1", which indicates that a previous reception has concluded. The microcontroller will be interrupted if the corresponding bit in the Interrupt Mask Register is set to logic "1". Otherwise, the microcontroller must periodically check the Status Register. Once the microcontroller is alerted to the fact that the previous reception has concluded, it may issue the "Enable Receive to Page fnn" command, which resets the RI bit to logic "0" and selects a new page in the RAM buffer. Again, the appropriate buffer size is specified in the "Define Configuration" command. Typically, the page which just received the data packet will be read by the microcontroller at this point.

Once the "Enable Receive to Page fnn" command is issued, the microcontroller attends to other duties. There is no way of knowing how long the new reception will take, since another node may transmit a packet at any time. When another node does transmit a packet to this node, and if the "Define Configuration" command has enabled the reception of long packets, the ARCNET core interprets the packet as either a long or short packet, depending on whether the content of the buffer location 2 is zero or non-zero. The format of the buffer is shown in Figure 12. Address 0 contains the Source Identifier (SID), Address 1 contains the Destination Identifier (DID), and Address 2 contains, for short packets, the value 256-N, where N represents the message length, or for long packets, the value 0, indicating that it is indeed a long packet. In the latter case, Address 3 contains the value 512-N, where N represents the message length. Note that on reception, the ARCNET core deposits packets into the RAM buffer in the same format that the transmitting node arranges them, which allows for a message to be received and then retransmitted without rearranging any bytes in the RAM buffer other than the SID and DID. Once the packet is received and stored correctly in the selected buffer, the ARCNET core sets the RI bit to logic "1" to signal the microcontroller that the reception is complete.

COMMAND CHAINING

The Command Chaining operation allows consecutive transmissions and receptions to occur without *on-chip 80C32* intervention. Through the use of a dual two-level FIFO, commands to be transmitted and received, as well as the status bits, are pipelined.

In order for the COM20051 to be compatible with previous SMSC ARCNET device *drivers,* the device defaults to the non-chaining mode. In order to take advantage of the Command Chaining operation, the Command Chaining Mode must be enabled via a logic "1" on bit 6 of the Configuration Register.

In Command Chaining, the Status Register appears as in Figure 12.

The following is a list of Command Chaining guidelines for the software programmer. Further detail can be found in the Transmit Command Chaining and Receive Command Chaining sections.

- The device is designed such that the interrupt service routine latency does not affect performance.
- Up to two outstanding transmissions and two outstanding receptions can be pending

at any given time. The commands may be given in any order.

- Up to two outstanding transmit interrupts and two outstanding receive interrupts are stored by the device, along with their respective status bits.

- The Interrupt Mask bits act on TTA (Rising Transition on Transmitter Available) for transmit operations and TRI (Rising Transition of Receiver Inhibited) for receive operations. TTA is set upon completion of a packet transmission only. TRI is set upon completion of a packet reception only. Typically there is no need to mask the TTA and TRI bits after clearing the interrupt.

- The traditional TA and RI bits are still available to reflect the present status of the device.

Transmit Command Chaining

When the microcontroller issues the first "Enable Transmit to Page fnn" command, the ARCNET core responds in the usual manner by resetting the TA and TMA bits to prepare for the transmission from the specified page. The TA bit can be used to see if there is currently a transmission pending, but the TA bit is really meant to be used in the non-chaining mode only. The TTA bits provide the relevant information for the device in the Command Chaining mode.

In the Command Chaining Mode, at any time after the first command is issued, the processor can issue a second "Enable Transmit from Page fnn" command. The ARCNET core stores the fact that the second transmit command was issued, along with the page number.

After the first transmission is completed, the ARCNET core updates the Status Register by setting the TTA bit, which generates an interrupt. The interrupt service routine should read the Status Register. At this point, the TTA bit will be found to be a logic "1" and the TMA (Transmit Message Acknowledge) bit will tell the processor whether the transmission was successful. After reading the Status Register, the "Clear Transmit Interrupt"

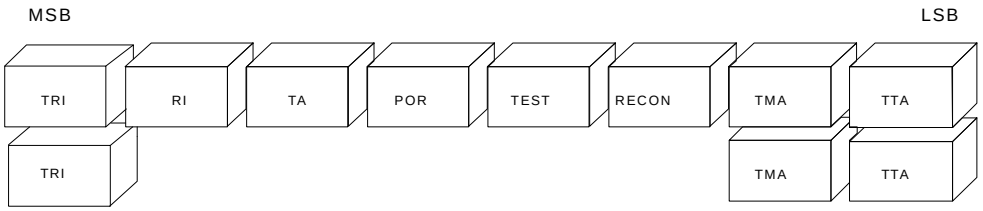


FIGURE 12 – COMMAND CHAINING STATUS REGISTER QUEUE

command is issued, thus resetting the TTA bit and clearing the interrupt. Note that only the "Clear Transmit Interrupt" command will clear the TTA bit and the interrupt. It is not necessary, however, to clear the bit or the interrupt right away because the status of the transmit operation is double buffered in order to retain the results of the first transmission for analysis by the processor. This information will remain in the Status Register until the "Clear Transmit Interrupt" command is issued. Note that the interrupt will remain active until the command is issued, and the second interrupt will not occur until the first interrupt is acknowledged. The ARCNET core guarantees a minimum of 200nS interrupt inactive time interval between interrupts. The TMA bit is also double buffered to reflect whether the appropriate transmission was a success. The TMA bit should only be considered valid after the corresponding TTA bit has been set to a logic "1". The TMA bit never causes an interrupt.

When the token is received again, the second transmission will be automatically initiated after the first is completed by using the stored "Enable Transmit from Page fnn" command. The operation is as if a new "Enable Transmit from Page fnn" command has just been issued. After the first Transmit status bits are cleared, the Status Register will again be updated with the results of the second transmission and a second interrupt resulting from the second transmission will occur. The ARCNET core guarantees a minimum of 200ns interrupt inactive time interval before the following edge.

The Transmitter Available (TA) bit of the Interrupt Mask Register now masks only the TTA bit of the Status Register, not the TA bit as in the non-chaining mode. Since the TTA bit is only set upon transmission of a packet (not by RESET), and since the TTA bit may easily be reset by issuing a "Clear Transmit Interrupt" command, there is no need to use the TA bit of the Interrupt Mask Register to mask interrupts generated by the TTA bit of the Status Register. In Command Chaining mode, the "Disable Transmitter" command will cancel the oldest transmission. This permits canceling a packet destined for a node not ready to receive. If both packets should be canceled, two "Disable Transmitter" commands should be issued.

Receive Command Chaining

Like the Transmit Command Chaining operation, the processor can issue two consecutive "Enable Receive from Page fnn" commands.

After the first packet is received into the first specified page, the TRI bit of the Status Register will be set to logic "1", causing an interrupt. Again, the interrupt need not be serviced immediately. Typically, the interrupt service routine will read the Status Register. At this point, the RI bit will be found to be a logic "1". After reading the Status Register, the "Clear Receive Interrupt" command should be issued, thus resetting the TRI bit and clearing the interrupt. Note that only the "Clear Receive Interrupt" command will clear the TRI bit and the interrupt. It is not necessary, however, to clear the bit or the interrupt right away because the status of the receive operation is double buffered in order to retain the results of the first reception for analysis by the processor, therefore the information will remain in the Status Register until the "Clear Receive Interrupt" command is issued. Note that the interrupt will remain active until the "Clear Receive Interrupt" command is issued, and the second interrupt will be stored until the first interrupt is acknowledged. A minimum of 200nS interrupt inactive time interval between interrupts is guaranteed.

The second reception will occur as soon as a second packet is sent to the node, as long as the second "Enable Receive to Page fnn" command was issued. The operation is as if a new "Enable Receive to Page fnn" command has just been issued. After the first Receive status bits are cleared, the Status Register will again be updated with the results of the second reception and a second interrupt resulting from the second reception will occur.

In the ARCNET core, the Receive Inhibit (RI) bit of the Interrupt Mask Register now masks only the TRI bit of the Status Register, not the RI bit as in the non-chaining mode. Since the TRI bit is only set upon reception of a packet (not by RESET), and since the TRI bit may easily be reset by issuing a "Clear Receive Interrupt" command, there is no need to use the RI bit of the Interrupt Mask Register to mask interrupts generated by the TRI bit of the Status Register.

In Command Chaining mode, the "Disable Receiver" command will cancel the oldest reception, unless the reception has already begun. If both receptions should be canceled, two "Disable Receiver" commands should be issued.

RESET DETAILS

Internal Reset Logic

The ARCNET core supports two reset options; software and hardware reset. A software reset is generated when a logic "1" is written to bit 7 of the Configuration Register. The device remains in reset as long as this bit is set. The software reset does not affect the contents of the Address Pointer Registers, the Configuration Register, the IMR, or the Setup Register. A hardware reset occurs when a high signal is asserted on the nRESET input. The minimum reset pulse width is 3.2 s *(for 20 MHz core clock, 1.6 s for 40 MHz core clock)* . This pulse width is used by the internal digital filter, which filters short glitches to allow only valid resets to occur.

Upon reset, the transmitter portion of the device is disabled and the internal registers assume those states outlined in the Internal Registers section.

After the nRESET signal is removed the user may write to the internal registers. Since writing a non-zero value to the Node ID Register wakes up the ARCNET core, the Setup Register should be written before the Node ID Register. Once the Node ID Register is written to, the ARCNET core reads the value and executes two write cycles to the RAM buffer. Address 0 is written with the data D1H and address 1 is written with the Node ID. The data pattern D1H was chosen arbitrarily, and is meant to provide assurance of proper microsequencer operation.

INITIALIZATION SEQUENCE

When the ARCNET core is powered on the internal registers may be written to. Since writing a non-zero value to the Node ID Register wakes up the core, the Setup Register should be written to before the Node ID Register. Until a non-zero value is placed into the NID Register, no microcode is executed, no tokens are passed by this node, and no reconfigurations are generated by this node. Once a non-zero value is placed in the register, the core wakes up, but the node will not attempt to join the network until the TX Enable bit of the Configuration Register is set.

Before setting the TX Enable bit, the software may make some determinations. The software may first observe the Receive Activity and the Token Seen bits of the Diagnostic Status Register to verify the health of the receiver and the network.

Next, the uniqueness of the Node ID value placed in the Node ID Register is determined. The TX Enable bit should still be a logic "0" until it is ensured that the Node ID is unique. If this node ID already exists, the Duplicate ID bit of the Diagnostic Status Register is set after a maximum of 840mS (or 1680mS if the ET1 and ET2 bits are other than 1,1). To determine if another node on the network already has this ID, the ARCNET core compares the value in the Node ID Register with the DID's of the token, and determines whether there is a response to it. Once the Diagnostic Status Register is read, the

DUPID bit is cleared. The user may then attempt a new ID value, wait 840mS before checking the Duplicate ID bit, and repeat the process until a unique Node ID is found. At this point, the TX Enable bit may be set to allow the node to join the network. Once the node joins the network, a reconfiguration occurs, as usual, thus setting the MYRECON bit of the Diagnostic Status Register.

The Tentative ID Register may be used to build a network map of all the nodes on the network, even once the COM20051 has joined the network. Once a value is placed in the Tentative ID Register, the ARCNET core looks for a response to a token whose DID matches the Tentative ID Register. The software can record this information and continue placing Tentative ID values into the register to continue building the network map. A complete network map is only valid until nodes are added to or deleted from the network. Note that a node cannot detect the existence of the next logical node on the network when using the Tentative ID. To determine the next logical node, the software should read the Next ID Register.

IMPROVED DIAGNOSTICS

The COM20051 allows the user to better manage the operation of the network through the use of the internal Diagnostic Status Register.

A high level on the My Reconfiguration (MYRECON) bit indicates that the Token Reception Timer of this node expired, causing a reconfiguration by this node. After the Reconfiguration (RECON) bit of the Status Register interrupts the microcontroller, the interrupt service routine will typically read the MYRECON bit of the Diagnostic Status Register. Reading the Diagnostic Status Register resets the MYRECON bit. Successive occurrences of a logic "1" on the MYRECON bit indicates that a problem exists with this node. At that point, the transmitter should be disabled so that the entire network is not held down while the node is being evaluated.

The Duplicate ID (DUPID) bit is used before the node joins the network to ensure that another node with the same ID does not exist on the network. Once it is determined that the ID in the Node ID Register is unique, the software should write a logic "1" to bit 5 of the Configuration Register to enable the basic transmit function. This allows the node to join the network.

The Receive Activity (RCVACT) bit of the Diagnostic Status Register will be set to a logic "1" whenever activity (logic "1") is detected on the RXIN pin.

The Token Seen (TOKEN) bit is set to a logic "1" whenever any token has been seen on the network (except those tokens transmitted by this node).

The RCVACT and TOKEN bits may help the user to troubleshoot the network or the node. If unusual events are occurring on the network, the user may find it valuable to use the TXEN bit of the Configuration Register to qualify events. Different combinations of the RCVACT, TOKEN, and TXEN bits, as shown indicate different situations:

Normal Results:

RCVACT=1, TOKEN=1, TXEN=0: The node is not part of the network. The network is operating properly without this node.

RCVACT=1, TOKEN=1, TXEN=1: The node sees receive activity and sees the token. The basic transmit function is enabled. Network and node are operating properly.
MYRECON=0, DUPID=0, RCVACT=1, TXEN=0, TOKEN=1: Single node network.

Abnormal Results:

RCVACT=1, TOKEN=0, TXEN=X: The node sees receive activity, but does not see the token. Either no other nodes exist on the network, some type of data corruption exists, the media driver is malfunctioning, the topology is set up incorrectly, there is noise on the network, or a reconfiguration is occurring.

RCVACT=0, TOKEN=0, TXEN=1: No receive activity is seen and the basic transmit function is enabled. The transmitter and/or receiver are not functioning properly.

RCVACT=0, TOKEN=0, TXEN=0: No receive activity and basic transmit function disabled. This node is not connected to the network.

The Excessive NAK (EXCNAK) bit is used to replace a timeout function traditionally implemented in software. This function is necessary to limit the number of times a sender issues a FBE to a node with no available buffer. When the destination node replies to 128 FBEs with 128 NAKs or 4 FBEs with 4 NAKs, the EXCNAK bit of the sender is set, generating an interrupt. At this point the software may abandon the transmission via the "Disable Transmitter" command. This sets the TA bit to logic "1" when the node next receives the token, to allow a different transmission to occur. The timeout value for the EXNACK bit (128 or 4) is determined by the FOUR-NAKS bit on the Setup Register.

The user may choose to wait for more NAK's before disabling the transmitter by taking advantage of the wraparound counter of the EXCNAK bit. When the EXCNAK bit goes high, indicating 128 or 4 NAKs, the "POR Clear Flags" command may be issued to reset the bit so that it will go high again after another count of 128 or 4. The software may count the number of times the EXCNAK bit goes high, and once the final count is reached, the "Disable Transmitter" command may be issued.

The New Next ID bit permits the software to detect the withdrawal or addition of nodes to the network.

The Tentative ID bit allows the user to build a network map of those nodes existing on the network. This feature is useful because it minimizes the need for human intervention. When a value placed in the Tentative ID Register matches the Node ID of another node on the network, the TENTID bit is set, telling the software that this NODE ID *already exists on the network.* The software should periodically place values in the Tentative ID Register and monitor the New Next ID bit to maintain an updated network map.

COM20051 APPLICATIONS INFORMATION

PROGRAMMING THE COM20051 NETWORK CORE

The COM20051 ARCNET core is relatively simple to program. The ARCNET core was designed to allow the processor to remain in control of the node and to perform network flow control with a minimum of processor intervention. There are two methods of operating the ARCNET core: in Command Chaining and Non-Command Chaining mode. Command chaining mode permits the processor to pipeline up to two transmit and two receive commands. In normal Non-Command Chaining mode, the microcontroller must access the ARCNET core for each individual transmission and reception since the ARCNET core can only handle one transmit and one receive command at any given time. The Command Chaining mode permits the ARCNET core to handle up to two transmit and two receive commands at any given time. Receive and Transmit status bits are buffered for each command in Command Chaining mode.

The basic software approach can be divided into three parts: ARCNET initialization, Transmit Interrupt servicing, and Receive Interrupt servicing. Both Command and Non-Command Chaining examples will be illustrated. It is strongly suggested that an interrupt driven approach be used since polling methods can result in lost transmit opportunities and in lost receptions.

ARCNET Initialization

Basic COM20051 ARCNET core initialization consists of the following:

1) Programming of the Address Decode Register
2) Configuring the Operating Mode (Command or Non-Command Chaining, Backplane or Dipulse operation)
3) Programming the Interrupt Routing Register (IRR)
4) Programming the Data Rate (optional)
5) Enabling the Receive All mode (optional)
6) Obtaining a Node ID and programming in its value
7) Entering the Network
8) Issuing initial Receive commands

Figure 14 is a flowchart of a typical initialization procedure. Steps 1 through 5 are straightforward and do not require much explanation. Step 6, Selection of a Node ID can be somewhat more involved depending on the application. There are many methods of obtaining a unique node id for a particular node ranging from simply reading a switch to sophisticated software algorithms. Several methods will be discussed but the best method depends on the particular application.

Node ID Selection

The ARCNET Protocol is a token passing protocol that relies on each station to have a unique Node ID in order to permit proper network operation. Many methods will be discussed but it is up to the user to choose the right method for his application.

Method 1 - Hardware Switch Read

This is the simplest and the most common form of obtaining a Node ID for an individual node. It consists of adding an eight bit DIP switch and address decoder as shown in Figure 15. A simple read from the microcontroller will obtain the ID value.

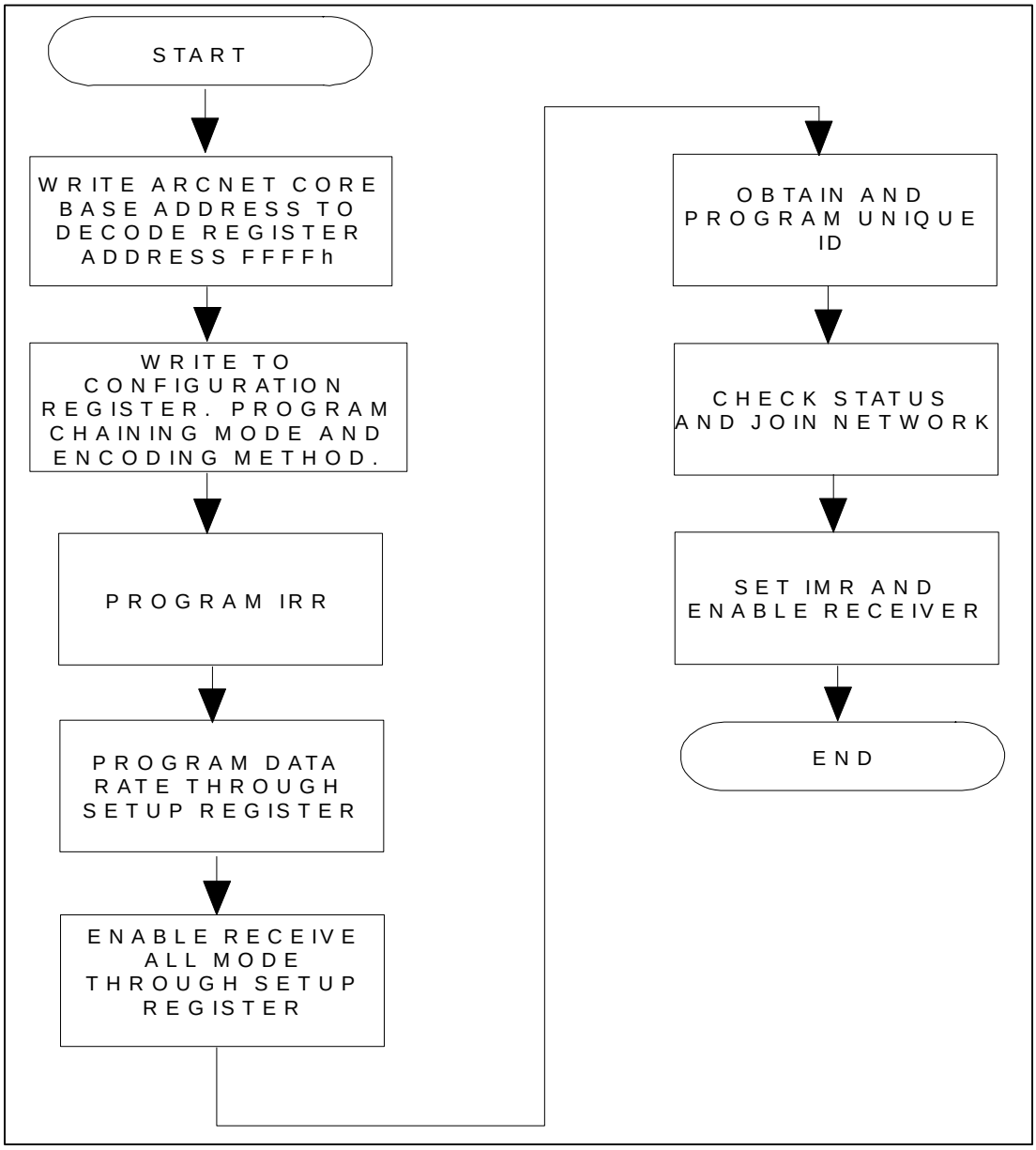


FIGURE 13 – TYPICAL INITIALIZATION PROCEDURE

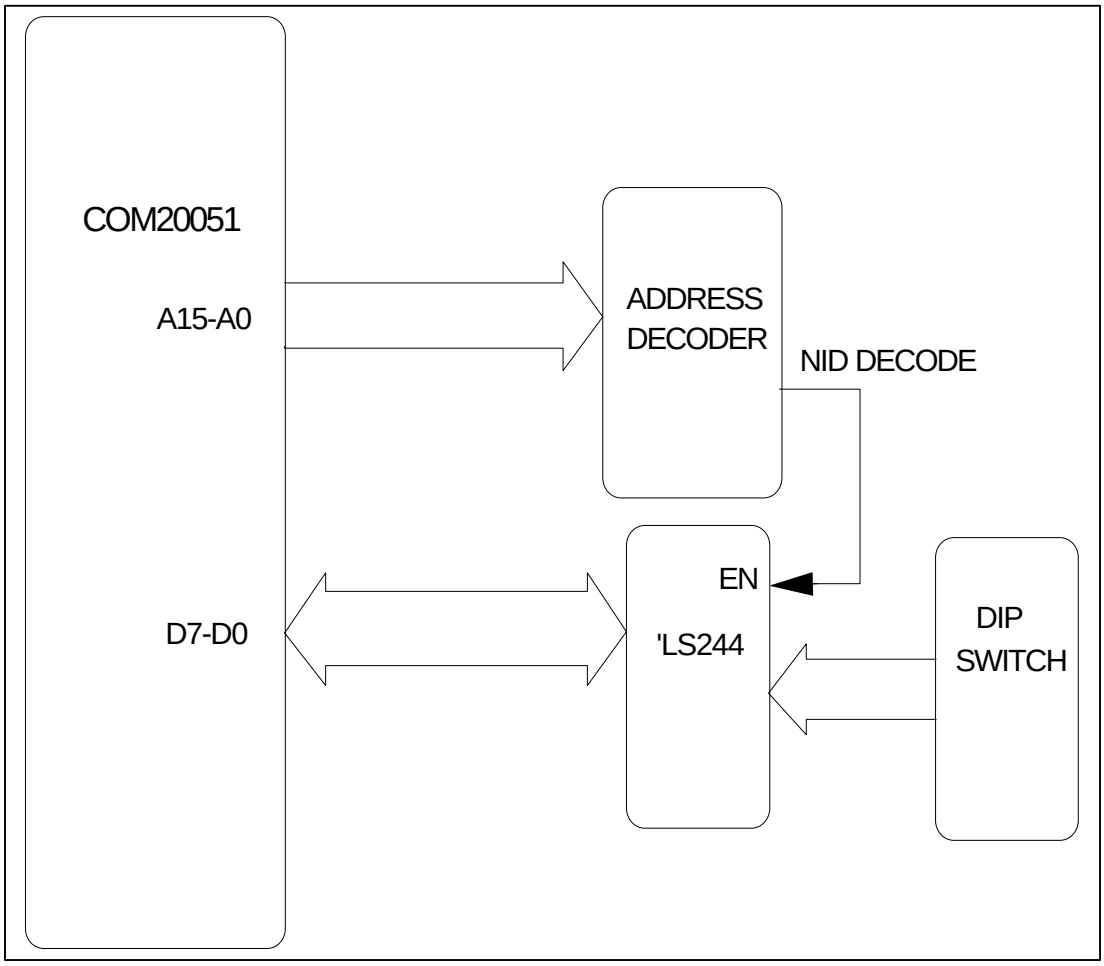


FIGURE 14 – NODE ID SELECTION USING A HARDWARE SWITCH READ

Method 2 - Non-Volatile Memory Storage

This method involves the storage of the Node ID in either a PROM, EPROM, or EEPROM. The microcontroller simply reads the proper location and programs the ID value. This method is well suited for closed networks that are not expected to expand or are maintained solely by the OEM. Such examples might include machine control and automobiles.

Method 3 - Hardwire Daisy Chain

The Daisy Chain method uses an additional wire running from node to node in a daisy chained fashion. This type of setup relies on a master node that uses a fixed ID value that is stored in the master node's non-volatile memory or by reading a switch. Once the master programs its ID value it brings the additional line low. The next node that is in line physically detects the low level on the line and uses a duplicate ID search algorithm to detect the first available ID on the

network. Once this node finds an ID and joins the network, it brings its ID line low to allow the next node to join. This process continues until all nodes have joined the network. Figure 16 shows an example of how this is accomplished.

Method 4 - Duplicate ID Software Search

The Duplicate ID software search is a software only method of locating available node ID values on a given network. The basic algorithm involves a sequential search of ID values and checking their availability by using a combination of diagnostic bits located within the ARCNET core of the COM20051. This method is not well suited for the initial power-up sequence of a network since there are instances in which the algorithm will assign the same ID to two nodes when they power-up simultaneously. This occurs due to basic constraints imposed by the ARCNET protocol itself. The Duplicate ID algorthim is very useful for adding nodes to already existing networks that use a mix of node ID selection methods. Please refer to the section on Duplicate ID Detection for details regarding the algorithm.

Entering the Network

After establishing a unique Node ID, the COM20051 can join the network. Prior to actually joining the network, several diagnostic bits should be checked to ensure that the node is properly functioning and that the network is functioning properly. The DUPID, RCVACT, and TOKEN bits should all be checked to ensure that the network is operating properly. A read of the Diagnostic Status register should be done to clear previous data prior to checking the status bits. This will ensure that the Diagnostic Status bits are reflecting the latest status of the network. A period of time lasting 840ms should elapse prior to checking the status bits. The 840ms period is a worst case situation based on a full 255 node network with every node transmitting a 512 byte packet. The actual wait time will be less for most systems and should be calculated based on the number of nodes and maximum packet size. Once the wait period is over, the Diagnostic Status bits should read as follows: DUPID = 0, RCVACT = 1, and TOKEN = 1. Any other combination indicates that a problem exists on the network. Once a valid status condition is established, the TXEN bit of the Configuration Regsiter should be programmed to a '1' to allow the transmitter output to drive the network. The node is now a member of the network.

Initializing Receptions and Interrupt Masking

Normally the receiver is enabled immediately after joining the network. Since a node never will know when a packet will be transmitted to it, it is advisable to have the receiver enabled at the earliest time possible. Generally, Enable to Receive commands are given at the end of the initialization. For systems using the Command Chaining feature, two receive commands should be given. For Non-Command Chaining systems,

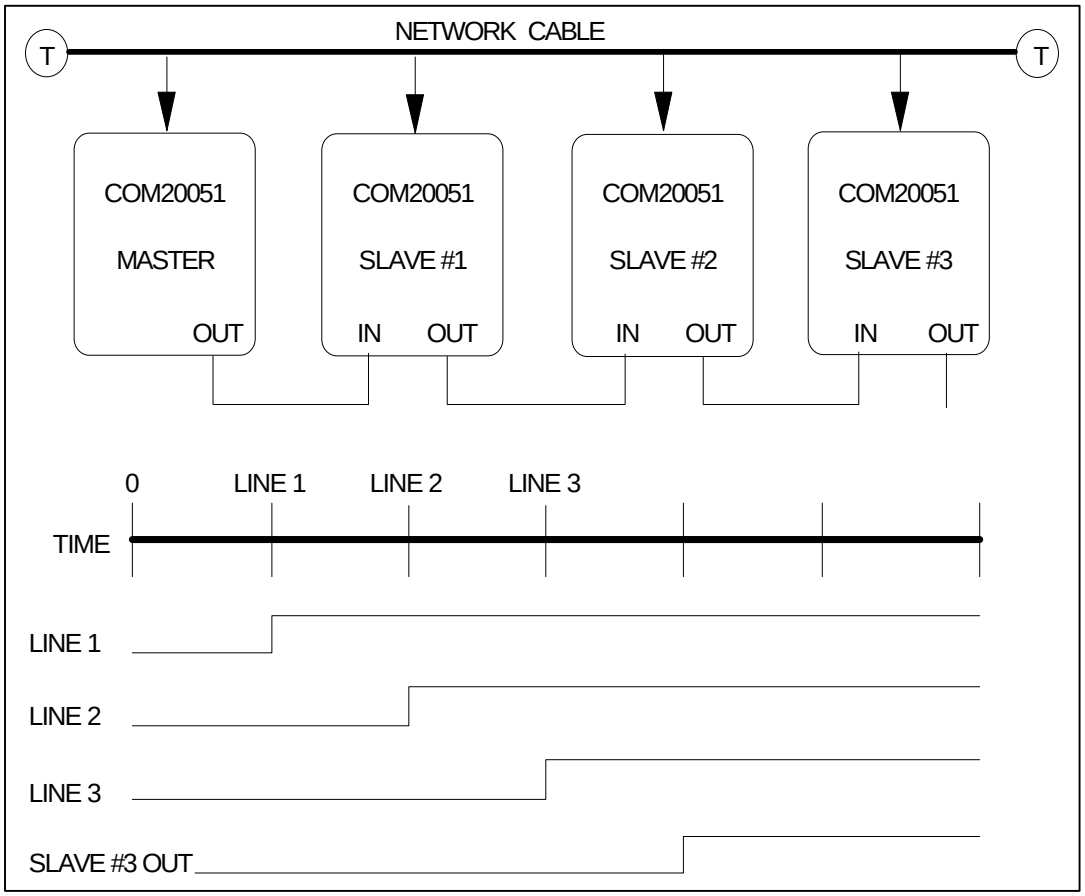


FIGURE 15 – DAISY CHAIN METHOD OF NODE ID SELECTION

a single Enable to Receive command should be issued.

Interrupt Masking is a critical function in Non-Command Chaining systems. The interrupt mask is critical because it provides the only method of releasing interrupts in Non-Command Chaining systems. For Non-Command Chaining systems, the RI interrupt should be unmasked right after the Enable to Receive commands to ensure that an interrupt will be generated. Note that the interrupt must be masked to release the interrupt once the packet has been received. For Command Chaining systems, the Receive Interrupt mask should be enabled and left enabled prior to the first Enable to Receive Commands. The Command Chaining mode utilizes a software Clear Receive Interrupt command to release the interrupt thus the interrupt can be left unmasked.

DUPLICATE ID DETECTION (AUTO NODE ID SELECTION)

The Duplicate ID algorithm was introduced previously as a software only method of obtaining unique ID values to join the network. The algorithm uses many of the diagnostic features found in the COM20051 including the RECON, RCVACT, TOKEN, DUPID, and MYRECON diagnostic bits to locate a unique ID value. Caution should be used when using this algorithm in that it cannot isolate a unique value when two nodes power-on simulatneously or within close proximity of each other. Two nodes using the same ID value will cause the network to fail.

The basic algorithm operates as follows: The COM20051's Node ID is temporaily initialized to FEh. A 52us waiting period is then entered to allow the core to read the new node ID value. This is necessary to allow time for the ARCNET microsequencer to read the value from the ARCNET Node register. A one second timing loop is then entered in which several diagnostic bits are sampled in order to determine if the selected node ID is being used or not. The first determination made is if the RCVACT (RECEIVER ACTIVITY) and the TOKEN bits are set. If both are reset, this node is most likely to be the first node to join the network. If one of these bits is reset, then a timing loop is entered and the bits are sampled again. Once both bits are found to be set then the DUPID bit is sampled. If this bit is set then the current node ID is decremented and the timing loop is restarted. If the DUPID bit is reset then the program continues in the timing loop and the RECON bit tested after the loop is exited to determine if a RECON had occurred during the sampling of the diagnostic status bits. If the RECON bit is reset then another loop is entered to synchronize the COM20051 with its Token Rotation timer in order to prevent multiple nodes from using the same Node ID. This loop samples the MYRECON bit which only gets set when the node has not seen a token to itself for 840ms. While the program is sampling the MYRECON bit, the DUPID bit is checked to prevent two nodes which have reconfigured close together from using the same ID. If the DUPID bit is set before the MYRECON bit is set then another node ID must be found. Once the MYRECON bit is found to be set then the TOKEN bit is checked again to make sure that another node has not RECONed just prior to this node. If all these conditions are satisfied then the TXEN bit of the Configuration Register is set and a transmission is sent out to the same Node ID. The TA and TMA bits are polled for 840ms. If the TMA bit is sampled as set within 840ms then the node ID is not valid and the selection process starts again. In the event that the MYRECON bit is not found to be set, an 840ms timer has been incorporated into the polling loop to account for this. The only time the MYRECON bit will not get set is when there is a single node network. Once a valid node ID has been located then the transmitter is enabled and the initialization process is completed.

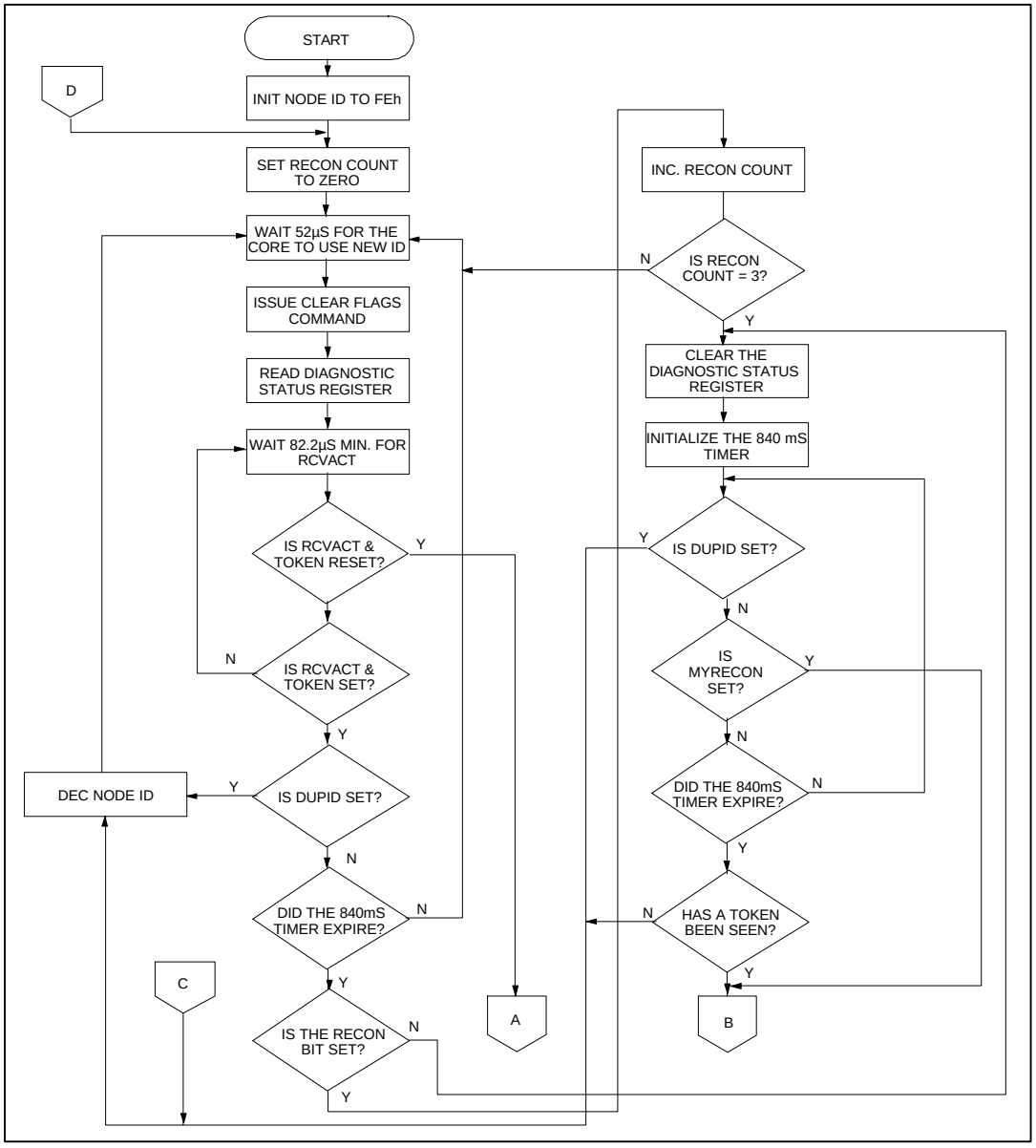


FIGURE 16 – DUPLICATE ID DETECTION ALGORITHM

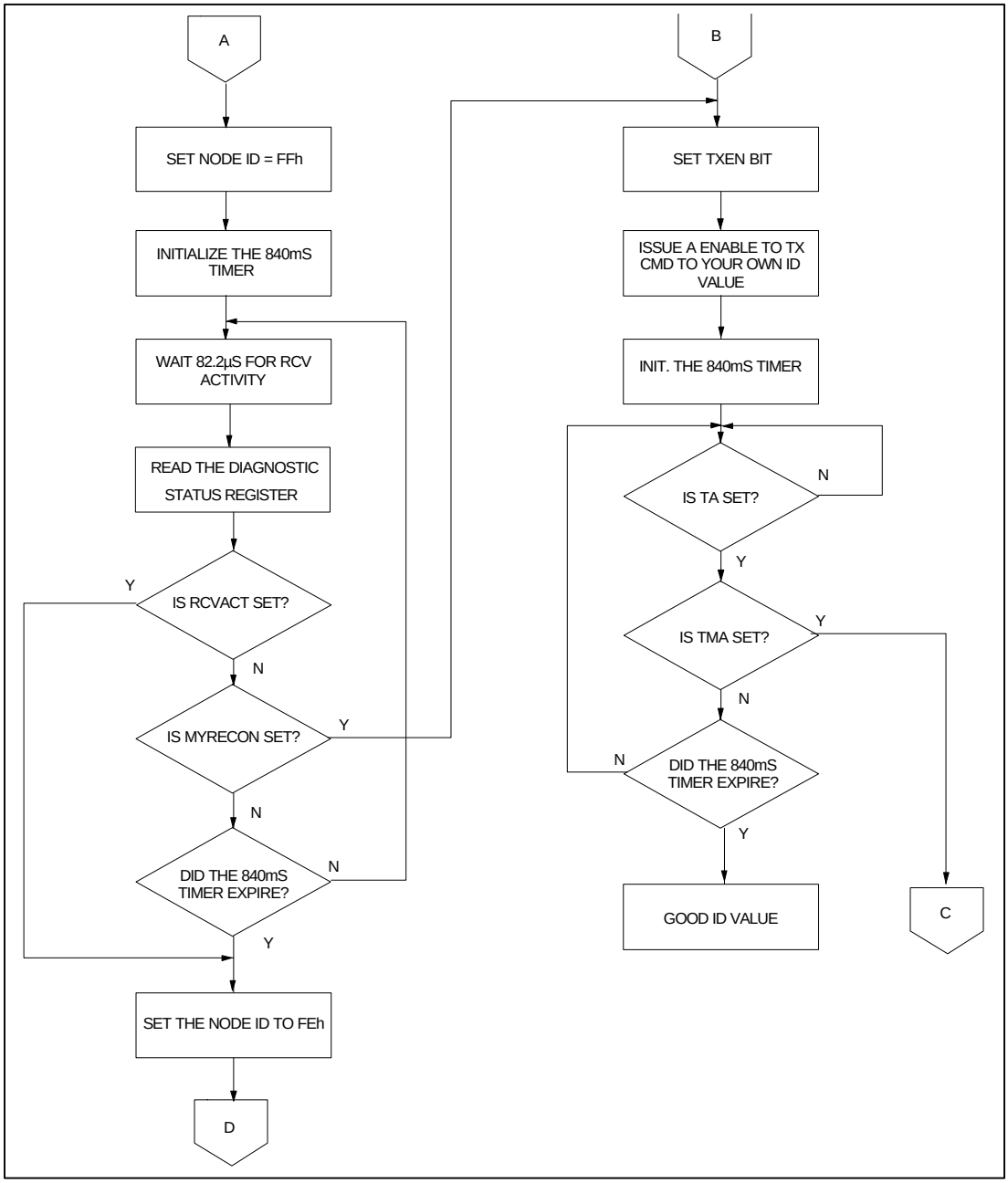


FIGURE 16 - DUPLICATE ID DETECTION ALGORITHM (continued)

DUPID Algorithm Exception Handling

There are two cases that require special processing that the normal Duplicate ID algorithm does not handle. The first case is when the node is the first node to join the network. The second case is when the node is the second node to join the network.

When a node is the first to join a network it will detect no receive activity and detect no tokens (i.e. RCVACT = 0 and TOKEN = 0). If this is the case, a special routine is entered that monitors the MYRECON bits and constantly polls the RCVACT, TOKEN, and DUPID bits. If any of the RCVACT, TOKEN, or DUPID bits gets set prior to the MYRECON bit, then another node has entered the network just prior to this one. If the MYRECON bit sets and the RCVACT, TOKEN, and DUPID bits remain reset then the node takes the ID value of FFh. Node ID FFh is reserved for the first node to join the network.

The second case involves a node that is the second node to join the network. In this case the node will detect the presence of tokens and receive activity (RCVACT & TOKEN = 1), and no duplicate IDs (DUPID = 0). The algorithm will detect that RECON bit has set. This is because the only existing node on the network is undergoing constant reconfigurations because it cannot pass tokens to another node. In order to prevent an endless loop from occurring due to the RECON test failure, a count of RECON bit tests is kept. If three consecutive polls of the RECON bit are positive it can be safely assumed that this node is the second node to join the network and takes the ID value of FEh.

Basic Transmit and Receive Service Routines

Transmit and Receive Service routines are often found together as part of an single interrupt service routine used to service the ARCNET interrupt. The COM20051 provides a 1Kx8 RAM for buffering of both transmit and receive packets and accommodates two types of packet formats:

A short format which takes a maximum of 253 bytes of data and a long format which takes a maximum of 508 bytes of data. Most industrial/embedded applications use the short (256 byte) format, thus allowing the RAM to buffer four packets. The following examples will use a two packet buffer for receptions and a two packet buffer for transmissions. Examples are shown for both Command and Non-Command Chaining systems. Figure 18 shows how a typical program will run.

Upon entry into the service routine all relevant 8051 registers are pushed onto the stack including the PSW and DPTR. The state of the COM20051 address registers is also saved so that any procedures that were interrupted in the process of accessing the COM20051 RAM are not corrupted. The COM20051 status register is then read and stored in a variable. A series of case statements then check the bit settings and branch to the appropriate service routine.

Service Priority

The ARCNET core of the COM20051 can generate a single interrupt from multiple sources including reception of packets, transmission of packets, EXcessive NACK counts, RECONfigurations, and New Next ID generation. When testing status bits to determine the source of the interrupt it is best to use the following priority:

1) RI - Receiver Inhibited (packet received)
2) TA - Transmitter Available (packet transmitted)
3) EXNACK - Excessive Number of NACK's to FBE's
4) RECON - either a full or mini RECON was detected
5) New Next ID - the node to which this node is passing the token to has not responded to the token pass and this node has found a new id to pass the token to.

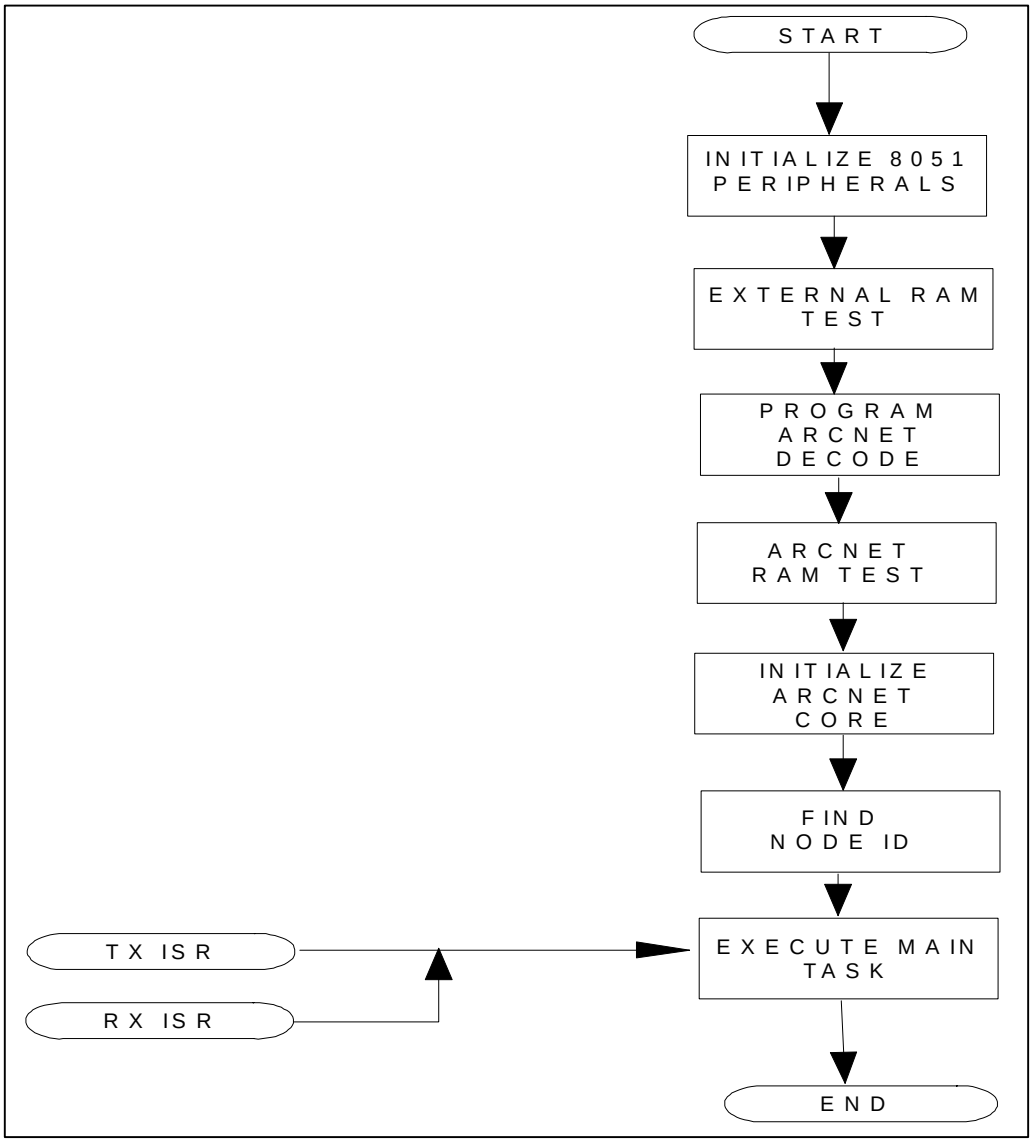


FIGURE 17 – TYPICAL PROGRAM EXECUTION

Receive Interrupt

A basic Receive service routine is shown in the flowchart of Figure 19. The basic concept of the service routine is not to process the packet data but to remove the packet from the ARCNET buffer and free the buffer for another reception. This method will minimize the number of missed packets due to unavailable buffers.

The Receiver Interrupt Service Routine (RX_ISR) will handle one reception at a time and enable another receive command upon exiting the service routine. Received data is stored in one of four buffers in external RAM. Several variables are kept in order to facilitate processing. RAM_BUF_REG is a byte wide, bit mapped software register which shows which external buffers are empty so that new data can be copied. RX_PAGE_REG is a byte wide register in which the lower nibble contains the ARCNET buffer RAM page address in which the latest reception can be found. The upper nibble contains the page address in which the next received is to be stored. This register allows the service routine to track from which page the most current data can be found when back to back Enable to Receive commands are issued. MIN_RX_PAGE and MAX_RX_PAGE are two constants that contain the page address boundaries for the ARCNET memory. If it is desired to allow more RAM for receptions and less RAM for transmissions then these constants must be changed accordingly.

Upon entering the RX_ISR the RX_PAGE_REG is read. The lower three bits are masked off and stored as the page address. RAM_BUF_REG is then read and the first free buffer found is used to store the packet. Once the data has been copied, the RAM_BUF_REG is checked to make sure that at least one buffer is empty before issuing another receive command. If no buffers are available, the interrupt is cleared and a new command is issued, thus discarding the received packet. This is done so that all receptions are handled. If it is found that receptions are being missed then more buffering on the CPU side is necessary. If a free buffer exists, the second receive command page is then checked to see if the maximum page address has been reached. If the page address is at its maximum value, then the page address is set to the minimum page address, otherwise the page address is incremented to next page. The Enable to Receive command string is then generated from the page address. RX_PAGE_REG is updated to make the second receive command the first command and the new command will be the second command. The actual command is then issued. The Clear Receive Interrupt command is then issued and the routine is exited.

Transmit Interrupt Servicing

The Transmit ISR (TX_ISR) services interrupts caused by the TA bit being set. The routine will first check the TMA bit to see if the last transmission has been received error free. If TMA is good then another transmission occurs. If TMA is bad then the last transmission is sent again and the ISR is exited. Limits can be imposed on the number of re-tries before aborting.

Several variables are kept in order to simplify servicing of the Transmit interrupt. TX_PEND_REG is a byte wide, bit mapped register that conveys information about whether a packet needs to be transmitted and where it is located in the ARCNET RAM. The lower nibble of the register tells the ISR from which external RAM page the transmit data is to come. Bits 5 and 6 are status bits telling the ISR how many transmit commands are in the Command Chaining pipeline. To identify from which ARCNET buffer page the last transmission originated, a variable called LAST_TX is used that contains the page number used in the last Enable to Transmit command. LAST_TX is used to re-transmit data when a bad TMA bit is found.

Upon entering the ISR, the status register is read and the TMA bit is checked. If the TMA bit is not set, then the action as described above is taken. If TMA is good then the interrupt is cleared and either bit 6 or bit 5 of the TX_PEND_REG is cleared depending on the number of transmissions pending. The configuration register is then read and the Command Chaining enable bit is checked. If Command Chaining is enabled, then bit 4 of TX_PEND is set, otherwise it is cleared. A series of case statements are then executed which read the TX_PEND_REG and look for a set bit. If a bit is set in the lower nibble, it signifies that the ARCNET RAM has been loaded with data and that page is ready to be transmitted. Software should be written so that an external routine will copy transmit data into the ARCNET RAM. This will speed up the ISR. Bits 5 and 6 are checked to see that the pipeline is not completely filled. If they are both set, then the routine is exited. If either one of the bits is reset, then the pending bit in TX_PEND_REG is reset, bit 5 or 6 is set, and an Enable to Transmit command is given to the page corresponding to the bit position in TX_PEND_REG.

Up to two transmit commands can be given within the ISR at any given time. If no transmissions are pending, then the routine is exited without any new transmissions.

Initiating Transmissions

The Transmit Service routine can only execute after a transmission occurs. Therefore, initial transmit commands must originate from routines external to the Transmit Interrupt service routine. The Transmit ISR can only transmit packets if and only if packets are pending transmission upon entry into the routine. Normally this is not the case, thus the programmer must take care to monitor and update the TX_PEND_REG for each transmission.

TX_PEND_REG

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
NOT USED	TX CMD #2 ISSUED	TX CMD #1 ISSUED	NOT USED	TX PEND. IN BUFFER 4	TX PEND. IN BUFFER 3	TX PEND. IN BUFFER 2	TX PEND. IN BUFFER 1

RAM_BUF_REG

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
NOT USED	NOT USED	NOT USED	NOT USED	NOT USED	NOT USED	RX BUF 2	RX BUF 1

RX_PAGE_REG

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
RECEIVE COMMAND PAGE #2				RECEIVE COMMAND PAGE #1			

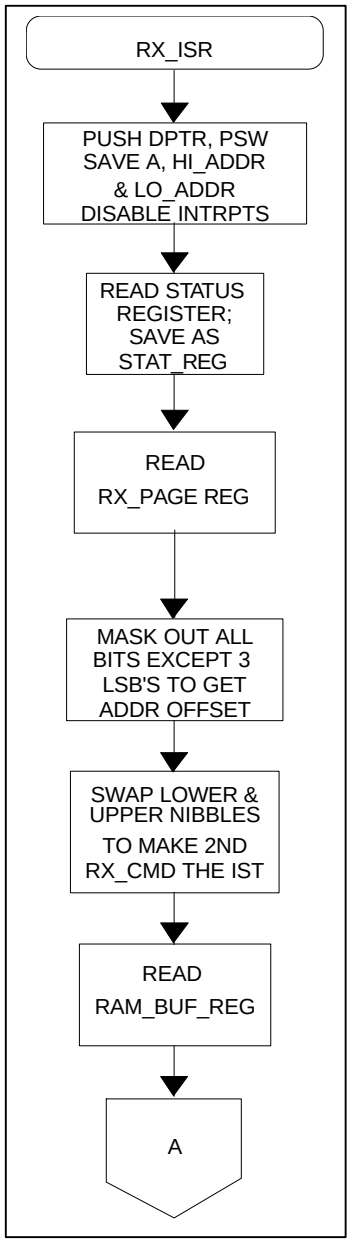


FIGURE 18 – RECEIVE INTERRUPT SERVICE ROUTINE

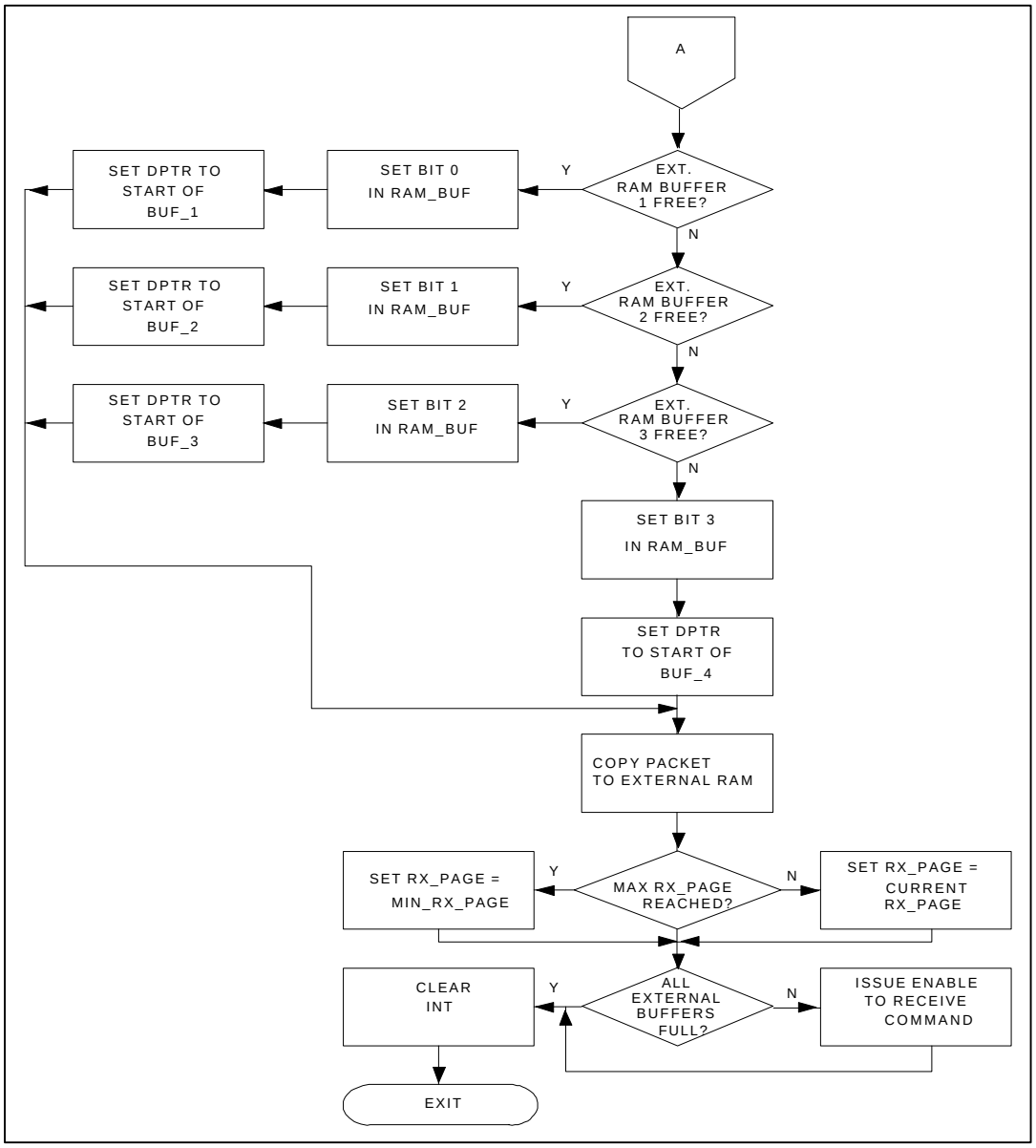


FIGURE 18 – RECEIVE INTERRUPT SERVICE ROUTINE (continued)

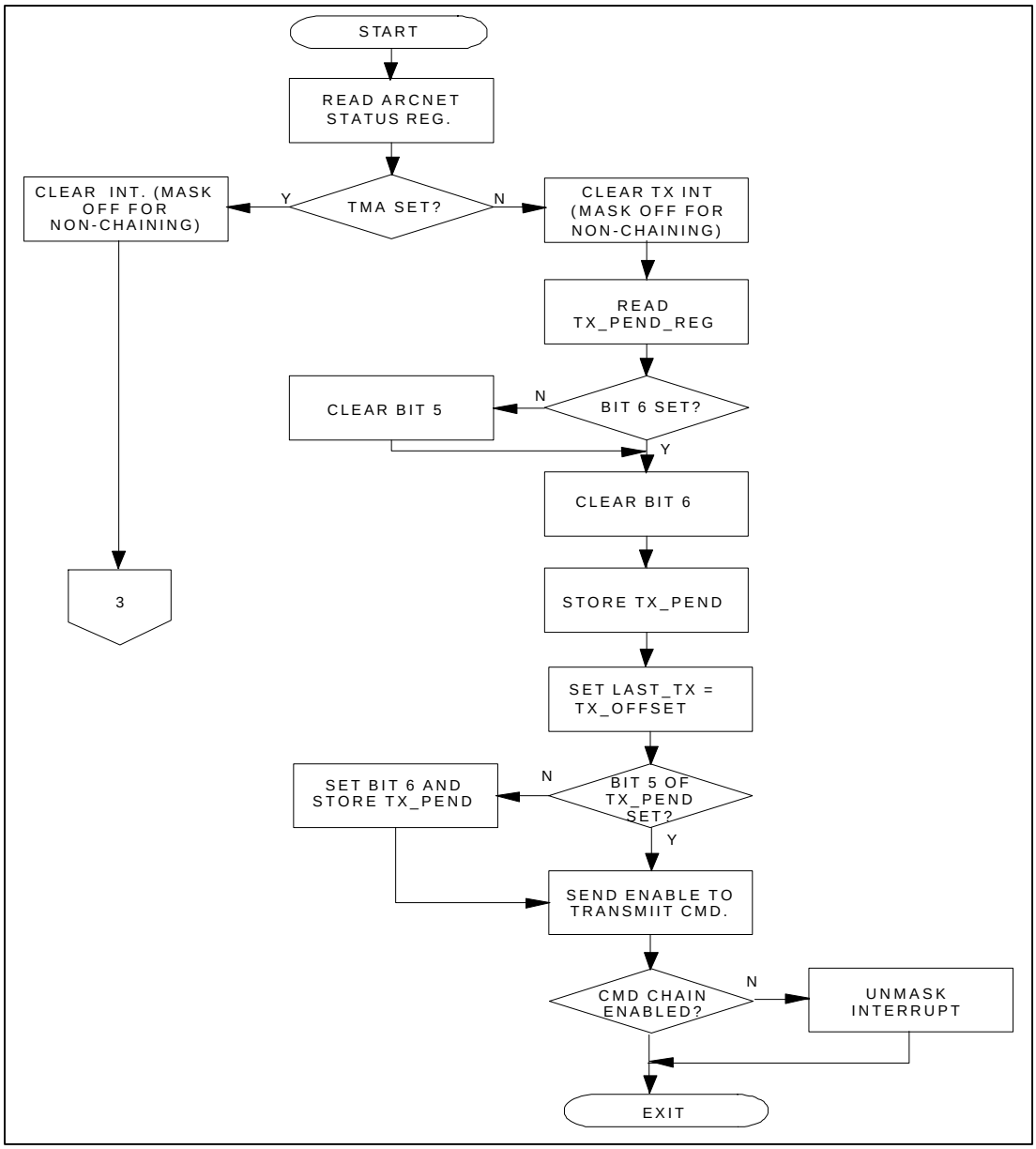


FIGURE 19 – TRANSMIT INTERRUPT SERVICE ROUTINE

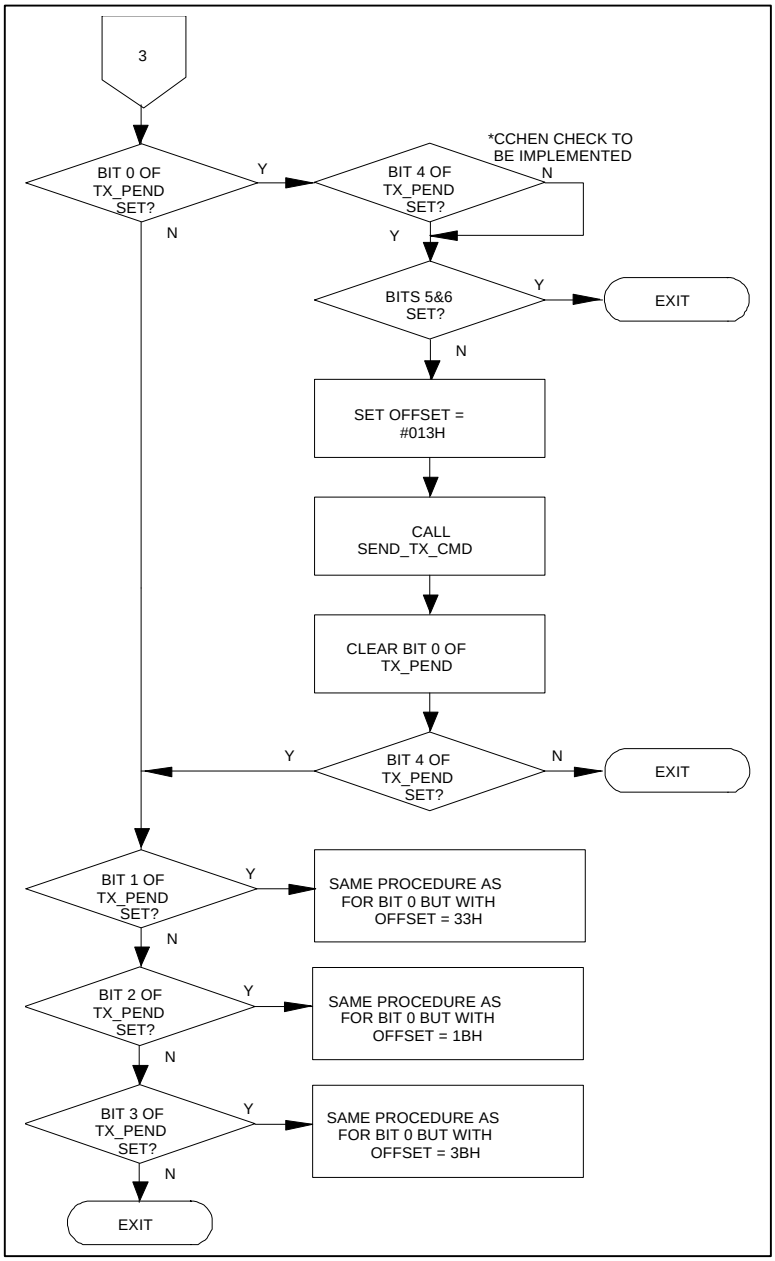


FIGURE 19 – TRANSMIT INTERRUPT SERVICE ROUTINE (continued)

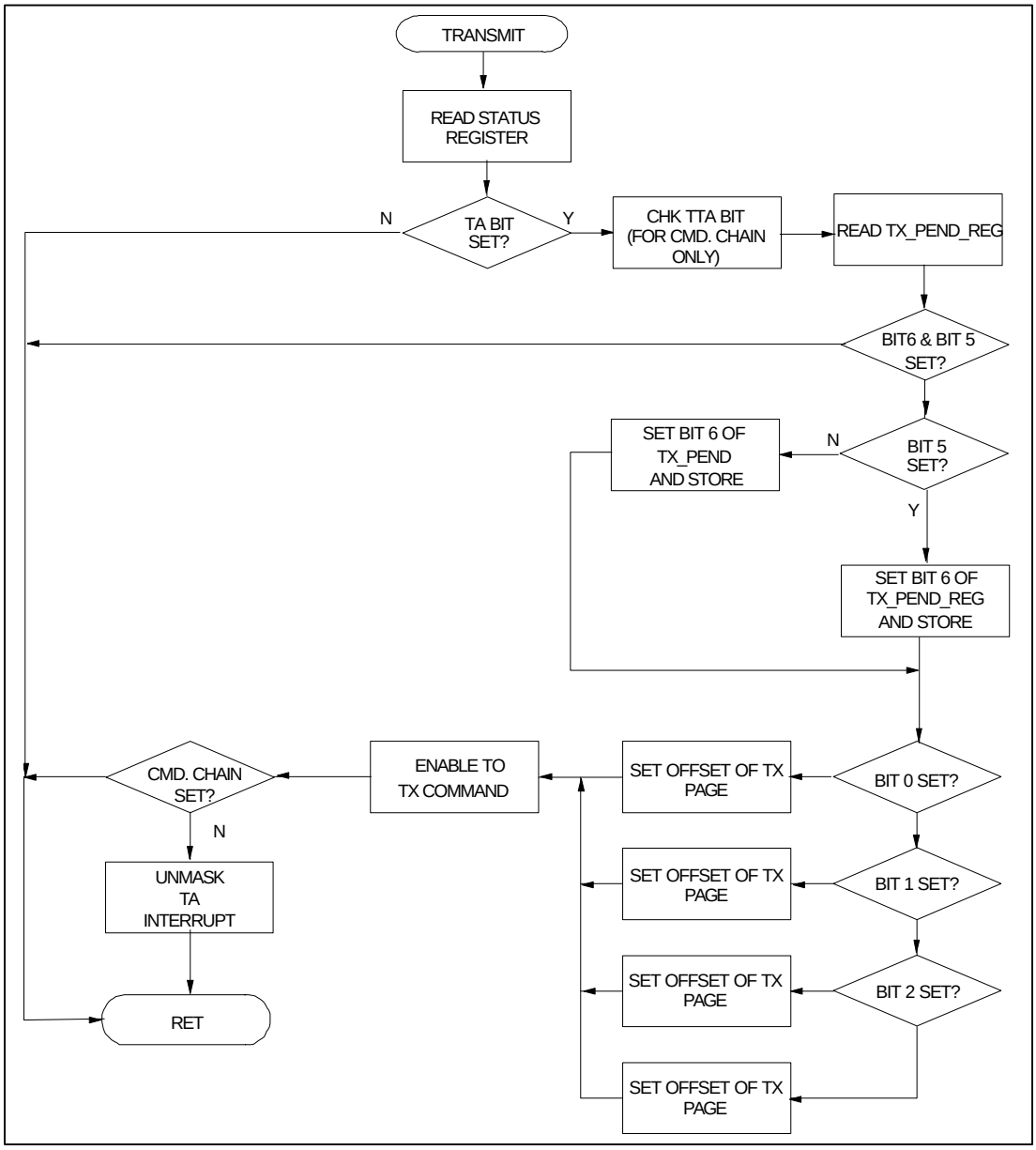


FIGURE 20 – EXTERNAL TRANSMIT ROUTINE

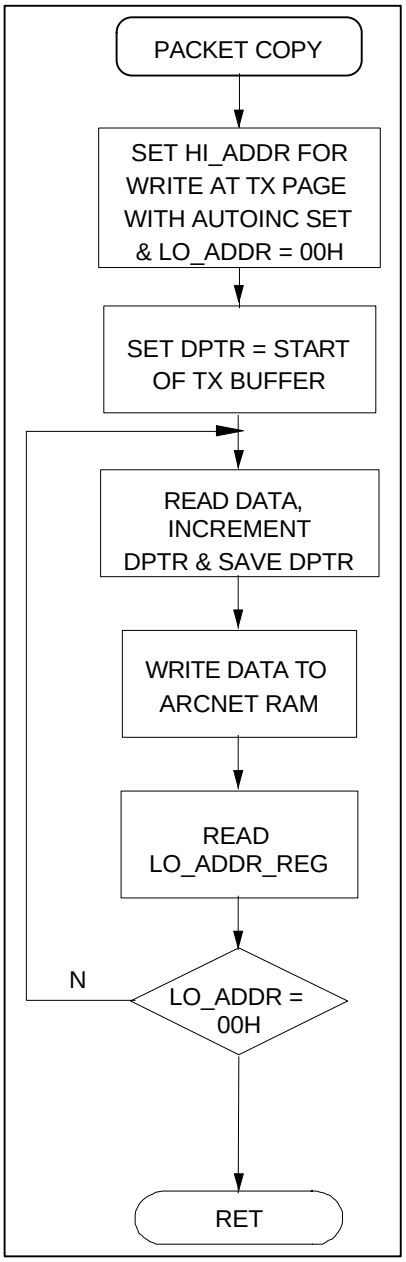


FIGURE 20 – EXTERNAL TRANSMIT ROUTINE (continued)

EXcessive Negative ACKnowledgement (EXNACK) Loops

Under certain conditions (node failure for example), a particular node's receiver may not become available. This will result in a NACK to FBE response each time the transmitter attempts to transmit a message. This can continue endlessly without CPU intervention and degrade network performance. In order to handle such failures, the COM20051 incorporates an EXNACK interrupt that will signal the CPU that it has made a specified number of Free Buffer requests which have not been acknowledged. The limit is specified by the 4NACKS bit of the Setup register of the ARCNET core. The default value is 128 re-tries but can be reduced to four by setting the 4NACKS bit. Once the limit is reached, an interrupt is generated by the ARCNET core. Note that FBEs will be issued until the CPU takes appropriate action by either aborting the transmission or continuing to retry.

The EXNACK (EXcessive Negative ACKnowledge) Interrupt handler is a simple procedure. For most applications, the Interrupt handler will abort the transmission by issuing a Disable Transmitter command. A Clear Flags command must be issued to release the interrupt and clear the EXNACK flag. Note that housekeeping should be done to signify to external procedures that the transmission was aborted and is still not pending. Figure 22 shows a typical flowchart of a EXNACK ISR.

Generating Network Maps

Most applications will require a table of existing nodes to be stored somewhere in the system for system administration and maintainence (it is usually the master node). The DUPID, TENTID, and New Next ID features are used to generate the Network Map. The map is generated by synchronizing the node to the token rotation time using the DUPID feature. Whenever a node is part of the network the DUPID bit will set when the node's receiver detects its own transmit activity in response to receipt of the token. Reading the Diagnostic Status register resets the DUPID bit. Resetting the bit and polling until the DUPID is set will synchronize the node the token rotation time. At this time a node ID value is programmed into the TENTID register. *Wait for the Token or the Recon bits to be set and clear Tent-ID bit. Then wait again for the Tent-ID bit to be set during one token rotation time.* This process should continure until all node ID values have been accounted for. There is one case in which the TENTID does not function. The TENTID detector cannot detect the node ID to which the node is passing the token. This is because the ARCNET core operates in a half-duplex mode thus blocking the recevier while it is transmitting the token. After compiling a node ID table using the TENTID feature, the Next ID register can be read to find out the missing ID value.

In order to keep the map updated, the New Next ID feature can be used to indicate changes in the network. The New Next ID feature will interrupt the processor any time the Next ID register is updated. This occurs whenever a node drops (causing a mini-recon) or when a node joins the network (causing a full recon). Whenever the Next ID interrupt is generated, a message should be sent to the Master node identifying a change in the network. The Master node can then update the network map accordingly.

Network Mapping While Off-Line

The previous mapping algorithm described a fast method of building a network map while on-line (being an active part of the network). Often it is desirable to build a map while off-line. A similar algorithm is used, but instead of using the TENTID feature the DUPID feature is used and the new node IDs are programmed into the node ID register. Since the node is not an active part of the network, there is no reliable method of

initially timing the token rotation. In order to find an existing ID to time the rotation, an 840ms timer is used. At the end of the 840ms, the DUPID is checked. If it is set then the node ID value exists and can be used as a fake ID (remember the node is not on the network). That first ID value remains in the node ID register and the above algorithm is used to determine the remainder of the ID values.

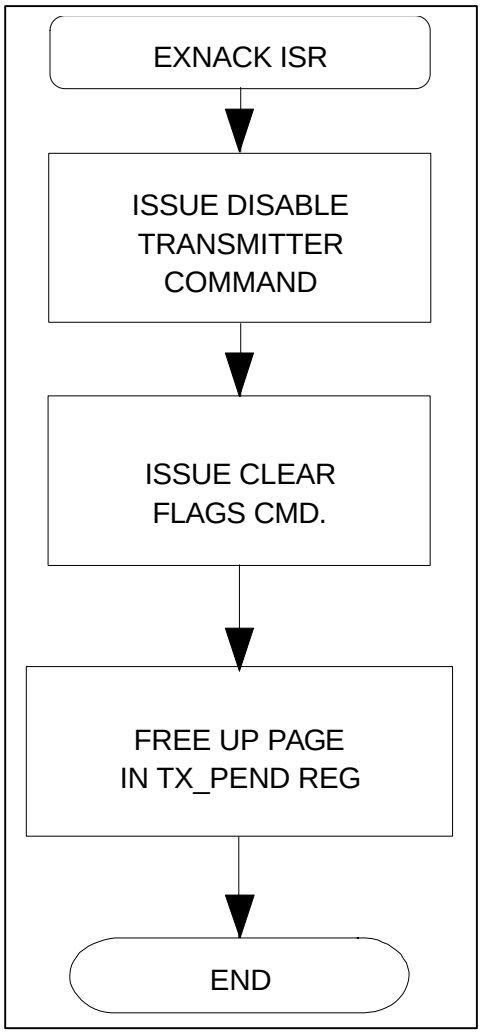


FIGURE 21 – EXCESSIVE NACK INTERRUPT SERVICE ROUTINE

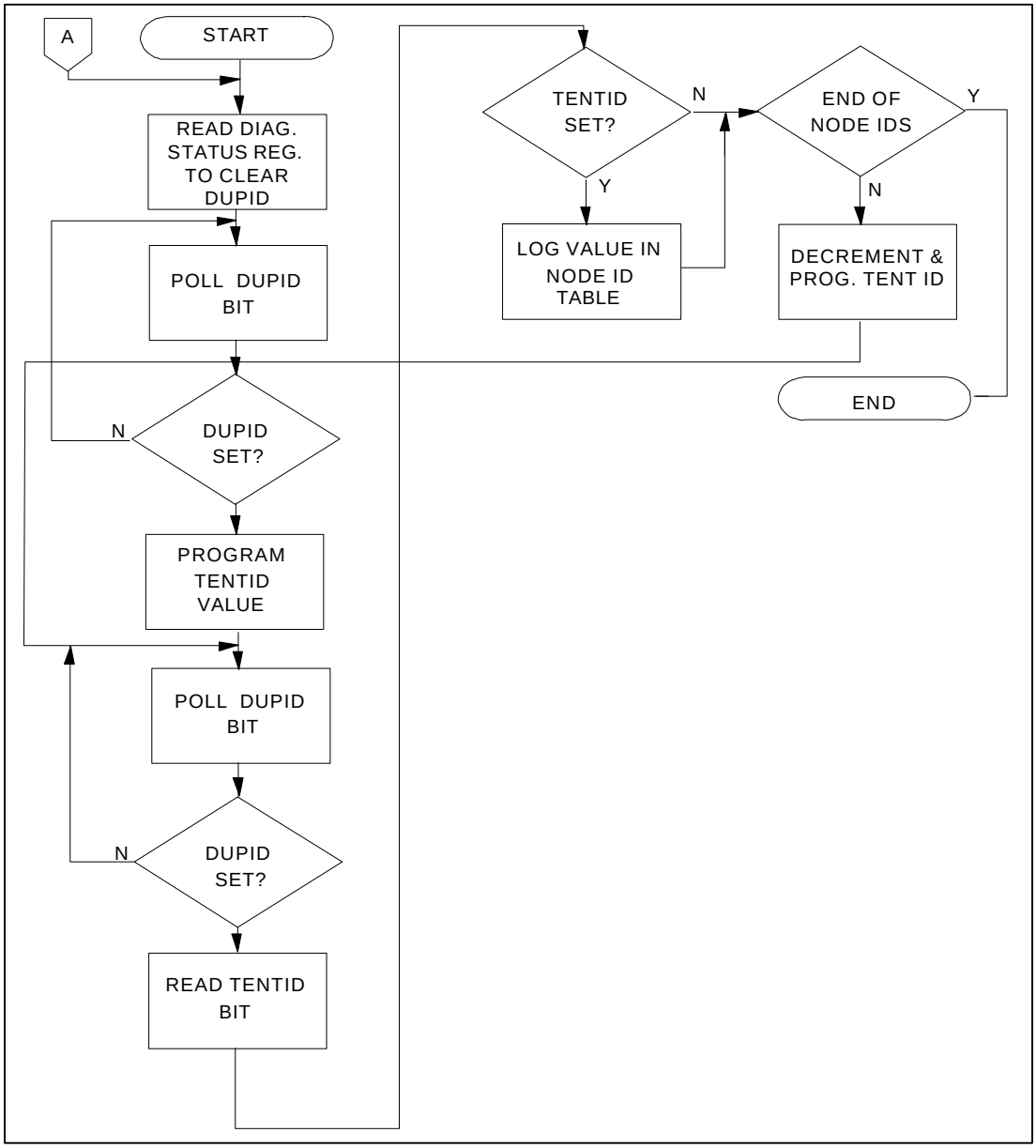


FIGURE 22A – ONLINE NETWORK MAPPING ROUTINE

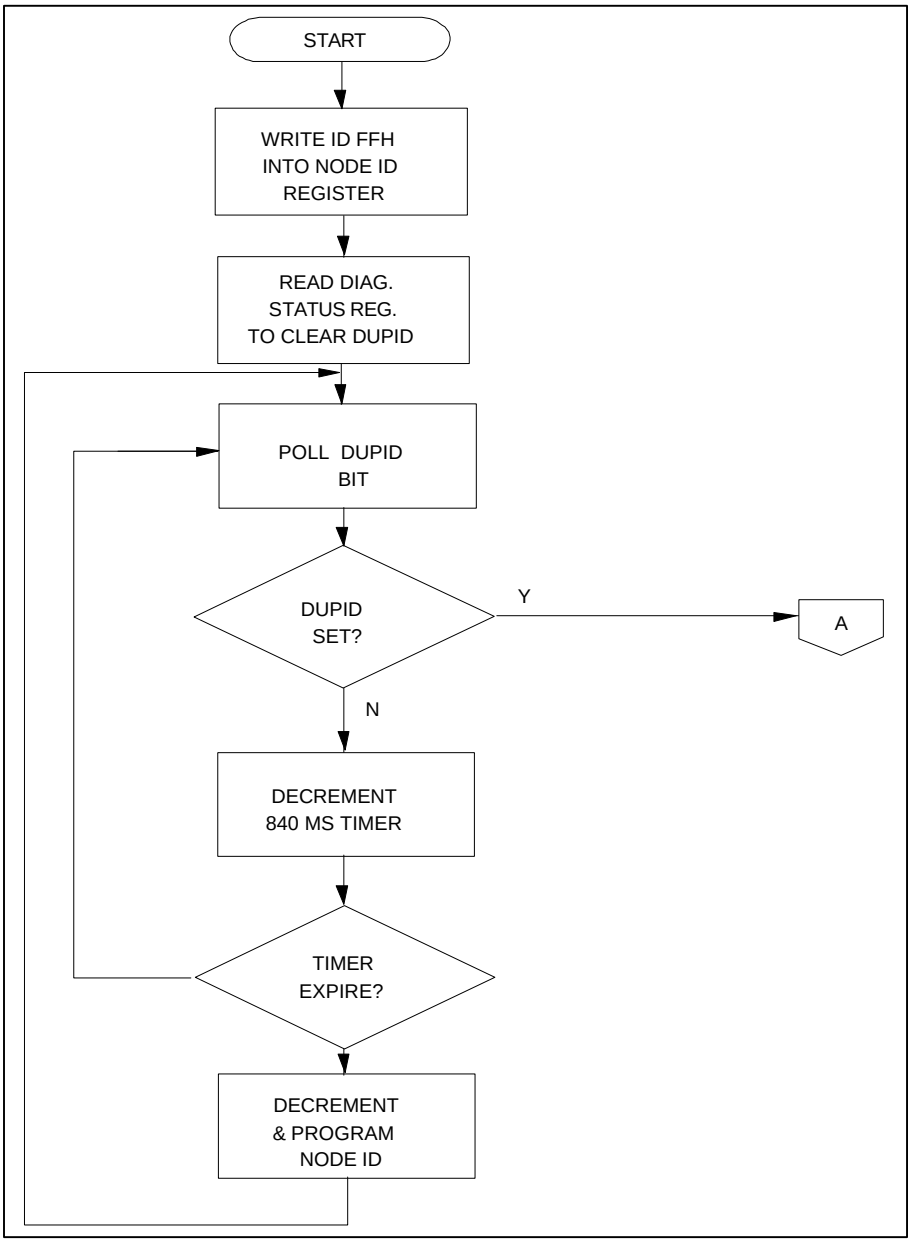


FIGURE 22B – ADDITIONAL CODE - OFFLINE NETWORK MAPPING ROUTINE

CABLING THE COM20051

The COM20051 supports several types of cables and cable interfaces in either bus or star topologies. The following is a list of media transceivers, supported media, and supported topologies:

TRANSCEIVER	ALLOWED TOPOLOGIES	MAX. DIST.	MAX. NODE	ISOLATION VOLTAGE	SUPPORTED MEDIA
HYC9068	Star only	2000ft.	N/A	<500V	Coax
HYC9088	Bus or Star	1000ft. (Coax) 400ft. (TP)	8 10	<500V	Coax Twisted Pair (TP)
RS-485	Bus	900ft.	32	15V D.C. Coupled	Twisted Pair
Isolated RS-485	Bus	700ft.	12	2.5KV	Twisted Pair
Fiber-Optic	Star	3Km	N/A	>2.5KV	Fiber

Note: The above figures are for a 2.5 Mbps data rate. Changing the data rate can significantly increase or decrease the line length and node count.

For more information please refer to TN7-5 RS-485 Cabling Guidelines for the COM20020 Universal Local Area Network Controller (ULANC) and Experimental Procedure for Verifcation of RS-485 Cabling Guidelines or Cabling Alternatives for the COM20020, COM90C66, and COM90C165.

Compatability Considerations

Compatability with existing ARCNET installations is of importance to some designers. The COM20051 is compatible with older ARCNET installations when it is operated in **dipulse mode at 2.5 Mbps using normal timeout values**. It is suggested that the HYC9068 or HYC9088 transceivers be used to maintain compatability. The COM20051 is not compatible with standard ARCNET systems when the device is used in backplane mode, when the reduced timeout features are enabled, or at data rates other than 2.5 Mbps.

USING THE COM20051'S EMULATION MODE

The COM20051 is unique among special purpose 8051 based devices in that it can be used in conjunction with a standard 16MHz 80C32 ICE. The COM20051 incorporates an Emulate mode that puts the internal microcontroller in a high impedance state so an external processor (like an ICE) can access the internal ARCNET core through the COM20051's pins. The ICE should be connected in parallel to the COM20051 as shown in Figure 24. The ICE should operate off of its internal oscillator.

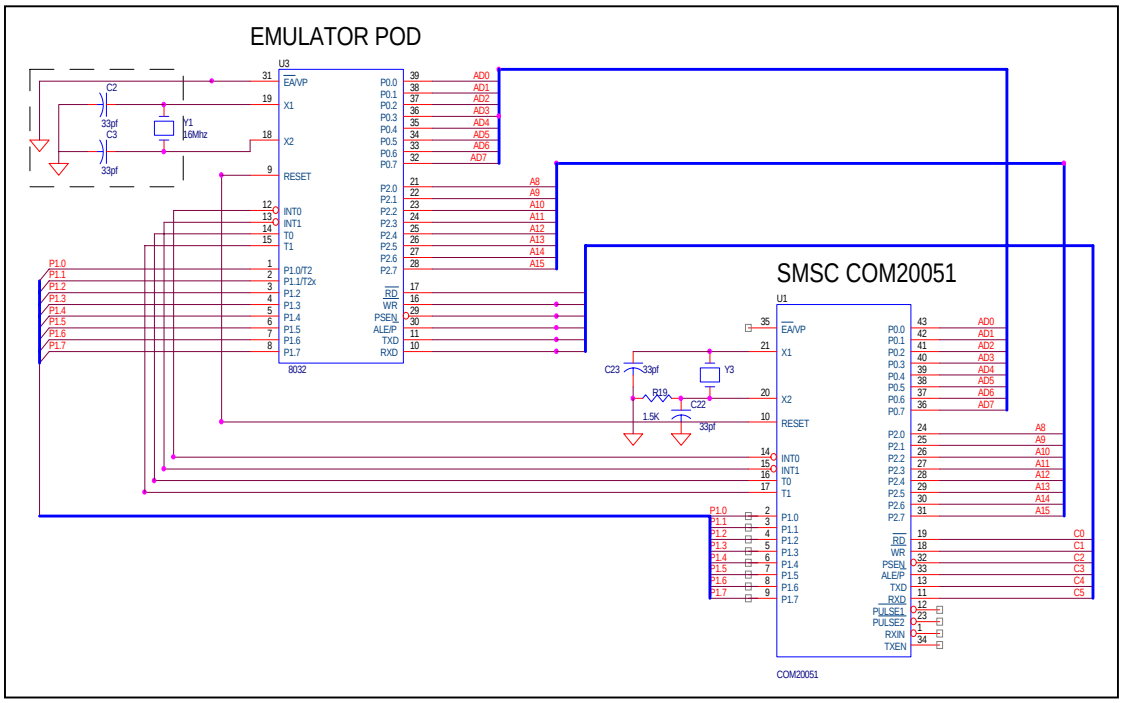


FIGURE 23 – CONNECTIONS FOR EMULATION MODE

OPERATIONAL DESCRIPTION

MAXIMUM GUARANTEED RATINGS*

Operating Temperature Range ..... $0^{o}C$ to $+70^{o}C$
Storage Temperature Range ..... $-55^{o}C$ to $+150^{o}C$
Lead Temperature (soldering, 10 seconds) ..... $+325\ ^{o}C$
Positive Voltage on any pin, with respect to ground ..... $V_{CC}+0.5V$
Negative Voltage on any pin, with respect to ground ..... -0.5V
Maximum V_{CC} ..... +7V

*Stresses above those listed may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other condition above those indicated in the operational sections of this specification is not implied.

NOTE: When powering this device from laboratory or system power supplies, it is important that the Absolute Maximum Ratings not be exceeded or device failure can result. Some power supplies exhibit voltage spikes or "glitches" on their outputs when the AC power is switched on or off. In addition, voltage transients on the AC power line may appear on the DC output. If this possibility exists it is suggested that a clamp circuit be used.

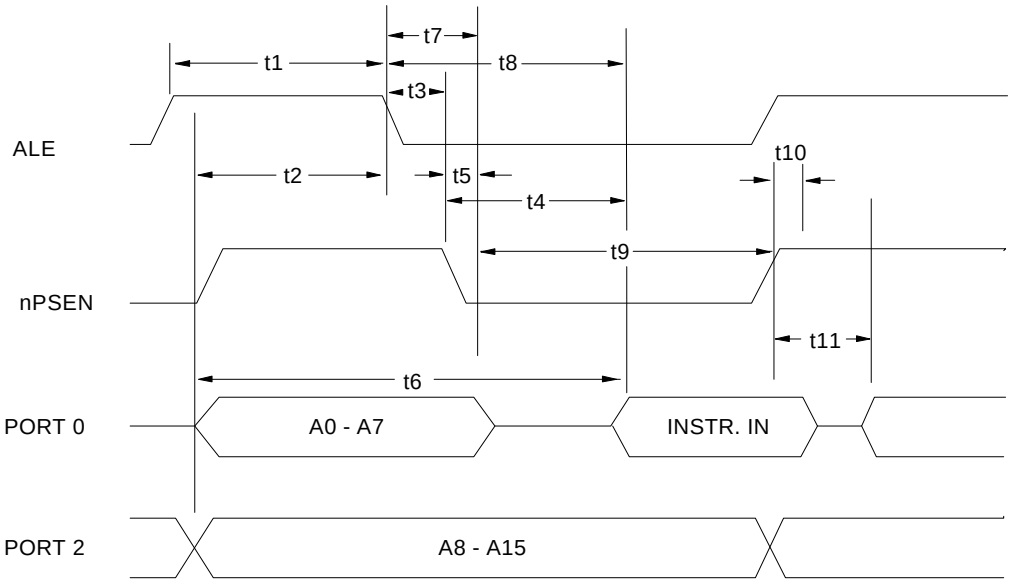
DC ELECTRICAL CHARACTERISTICS (T_A = 0 C - 70 C, V_{CC} = 5.0V ± 10%, V_{SS} = 0V)

PARAMETER	SYMBOL	MIN	TYP	MAX	UNIT	COMMENT
Low Output Voltage (Ports 1, 2 and 3, ALE, PSEN)	V_{OL}			0.3 0.45 1.0	V V V	I_{OL} = 100 A I_{OL} = 1.6 mA I_{OL} = 3.5 mA
Low Output Voltage (Port 0)	V_{OL1}			0.3 0.45 1.0	V V V	I_{OL} = 200 A I_{OL} = 3.2 mA I_{OL} = 7.0 mA
High Output Voltage (Ports 1, 2 and 3, ALE, PSEN)	V_{OH}	V_{CC}-0.3 V_{CC}-0.7 V_{CC}-1.5			V V V	I_{OH} = -10 A I_{OH} = -30 A I_{OH} = -60 A
High Output Voltage (Port 0 in External Bus Mode)	V_{OH1}	V_{CC}-0.3 V_{CC}-0.7 V_{CC}-1.5			V V V	I_{OH} = -200 A I_{OH} = -3.2 mA I_{OH} = -7.0 mA
Input Leakage Current	I_{LI}		.02 A	±10 µA	V	$0<V_{IN}<V_{CC}$
RST Pulldown Resistor	RRST	50K			Ω	
Pin Capacitance	CIO		10		pF	
Power Supply Current	I_{CC}			50	mA	
Low Input Voltage (XTAL1)	V_{IL2}			1.0	V	

PARAMETER	SYMBOL	MIN	TYP	MAX	UNIT	COMMENT
High Input Voltage	V_{IH2}	4.0			V	
Low to High Threshold Input Voltage (RESET, RXIN)	V_{ILH}		1.8		V	Schmitt Trigger @ 5V
High to Low Threshold Input Voltage (RESET, RXIN)	V_{IHL}		1.2		V	Schmitt Trigger @ 5V
Low Output Voltage (PULSE1 in Normal Mode, PULSE2, TXEN)	V_{OL3}			0.4	V	I_{SINK} = 4mA
High Output Voltage (PULSE1 in Normal Mode, PULSE2, TXEN)	V_{OH3}	2.4			V	I_{SOURCE} = 2mA
Low Output Voltage (PULSE1 in Backplane Mode)	V_{OL4}			0.5	V	I_{SINK} = 48mA
Output Capacitance (PULSE1 in Backplane Mode)	C_{OUT2}			400	pF	

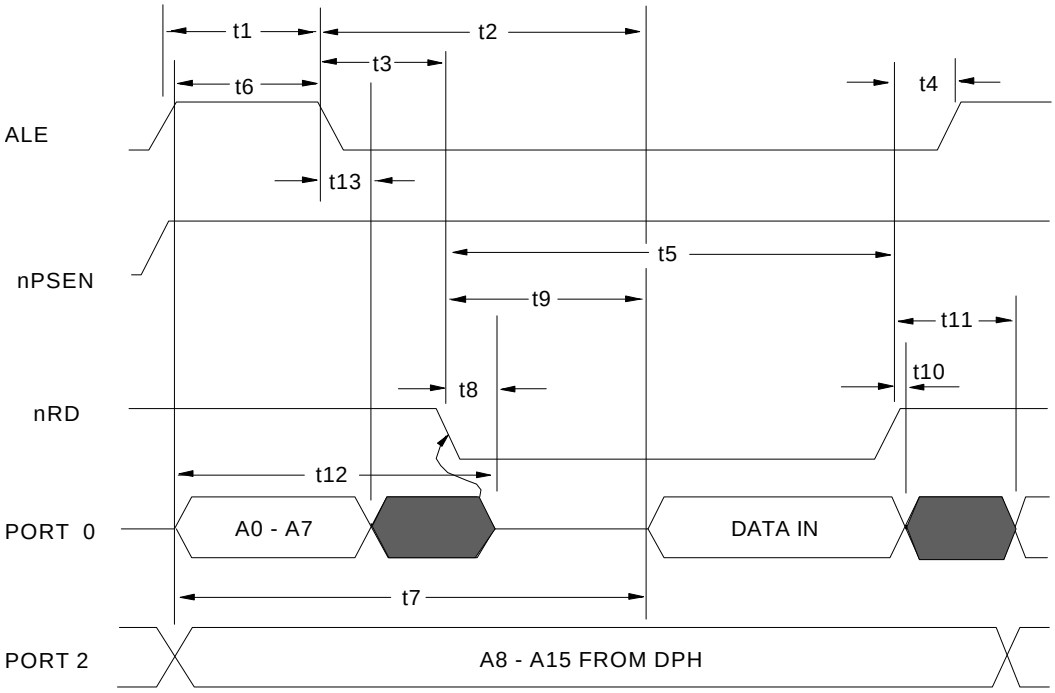
TIMING DIAGRAMS

AC CHARACTERISTICS (0°C - 70°C, 5.0V ± 10%)
100 pF Loading for ALE, PORT0, and nPSEN; 16 MHz Processor Clock



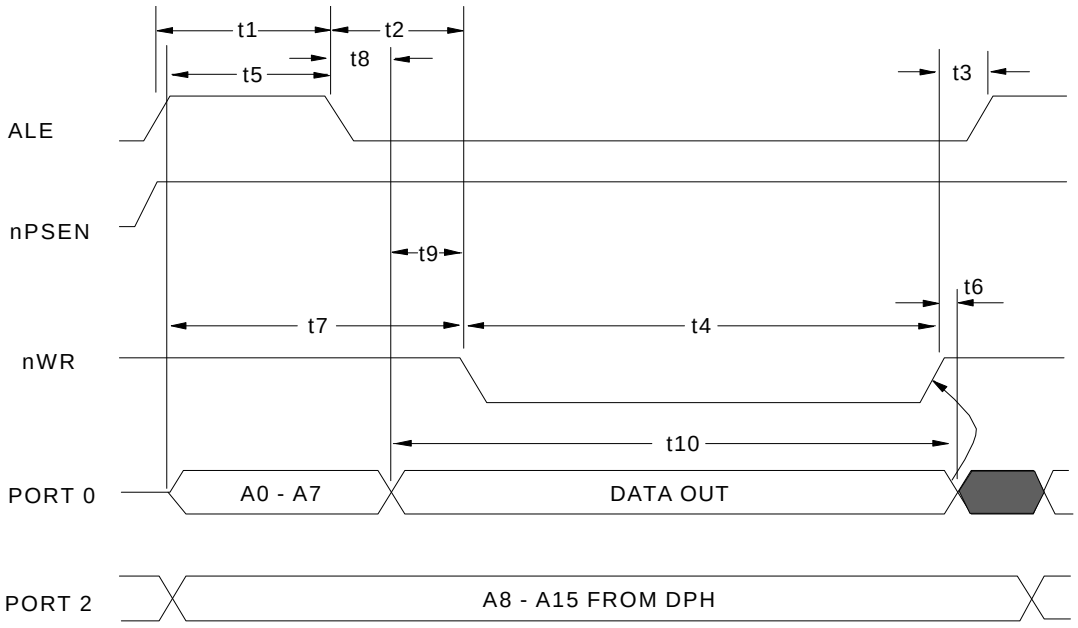
	Parameter	min	typ	max	units
t1	ALE Pulse Width	85			nS
t2	Address Valid to ALE Low	7			nS
t3	ALE Low to nPSEN Low	22.6			nS
t4	nPSEN Low to Valid Instruction Input			82.5	nS
t5	nPSEN Low to Address Float			10	nS
t6	Address to Valid Instruction Input			207.5	nS
t7	Address Hold after ALE	27			nS
t8	ALE Low to Valid Instruction Input			150	nS
t9	nPSEN Pulse Width	142.5			nS
t10	Input Instruction Hold after nPSEN	0			nS
t11	Input Instruction Float after nPSEN			37.5	nS

FIGURE 24 – EXTERNAL PROGRAM MEMORY READ CYCLE



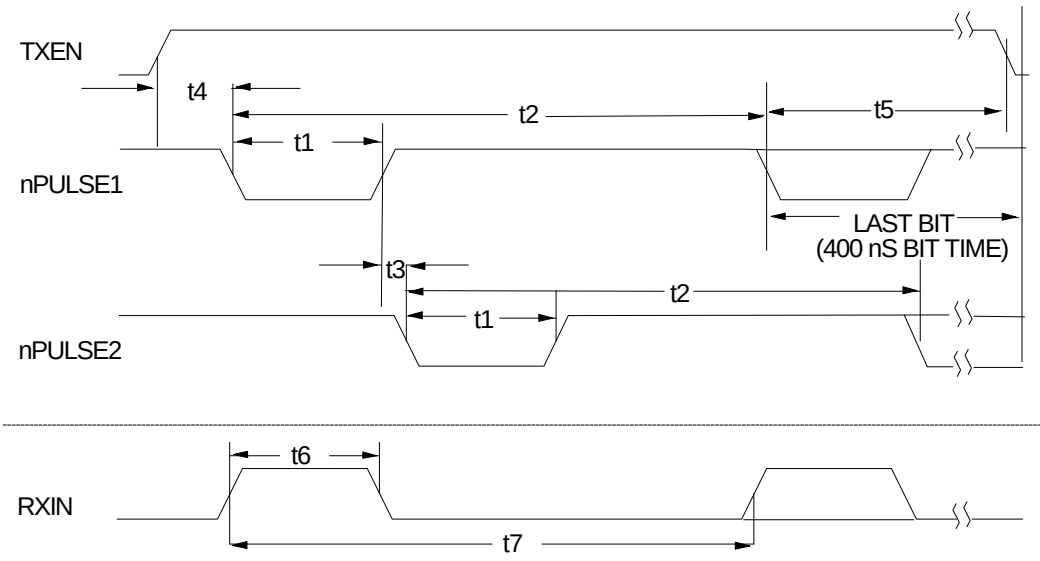
	Parameter	min	typ	max	units
t1	ALE Pulse Width	85			nS
t2	ALE Low to Valid Data In			350	nS
t3	ALE Low to nRD or nWR Low	137.5		238.5	nS
t4	nRD or nWR High to ALE High	22.5		102.5	nS
t5	nRD Pulse Width	275			nS
t6	Address Valid to ALE Low	7			nS
t7	Address to Valid Data In			397.5	nS
t8	nRD Low to Address Float			0	nS
t9	nRD Low to Valid Data In			147.5	nS
t10	Data Hold after nRD	0			nS
t11	Data Float after nRD			55	nS
t12	Address Valid to nRD or nWR Low	120			nS
t13	Address Hold after ALE Low	27			nS

FIGURE 25 – EXTERNAL DATA MEMORY READ CYCLE



	Parameter	min	typ	max	units
t1	ALE Pulse Width	85			nS
t2	ALE Low to nRD or nWR Low	137.5		238.5	nS
t3	nRD or nWR High to ALE High	22.5		102.5	nS
t4	WR Pulse Width	275			nS
t5	Address Valid to ALE Low	7			nS
t6	Data Hold after nWR	12.5			nS
t7	Address Valid to nRD or nWR Low	120			nS
t8	Address Hold after ALE Low	27			nS
t9	Data Valid to nWR Transition	2			nS
t10	Data Valid to nWR High	287.5			nS

FIGURE 26 – EXTERNAL DATA MEMORY WRITE CYCLE



	Parameter	min	typ	max	units
t1	PULSE1, PULSE2 Pulse Width		100*		nS
t2	PULSE1, PULSE2 Period		400*		nS
t3	PULSE1, PULSE2 Overlap	-10	0	+10	nS
t4	TXEN Low to PULSE1 Low**	850		950	nS
t5	Beginning of Last Bit Time to TXEN High**	250		350	nS
t6	RXIN Pulse Width	10	100*		nS
t7	RXIN Period		400*		nS

* t1, t6 = .25 x ($\frac{1}{\text{data rate}}$) for data rates other than 2.5 Mpbs.

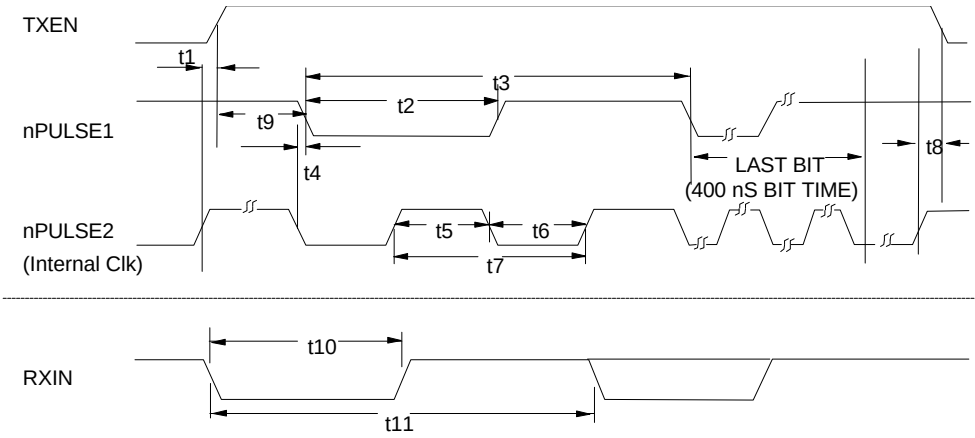
* t2,t7 = 8 x (crystal period) for clock frequencies other than 20 MHz. This period applies to data of two consecutive one's.

** t4: For clock frequencies other than 20 MHz, t4 = 18 x (crystal period) ± 50 nsec.

** t5: For clock frequencies other than 20 MHz, t5 = 6 x (crystal period) ± 50 nsec.

Note: Clock frequency for 5 Mbps is 40 MHz.

FIGURE 27 – NORMAL MODE TRANSMIT OR RECEIVE TIMING
(These signals are to and from the Hybrid)

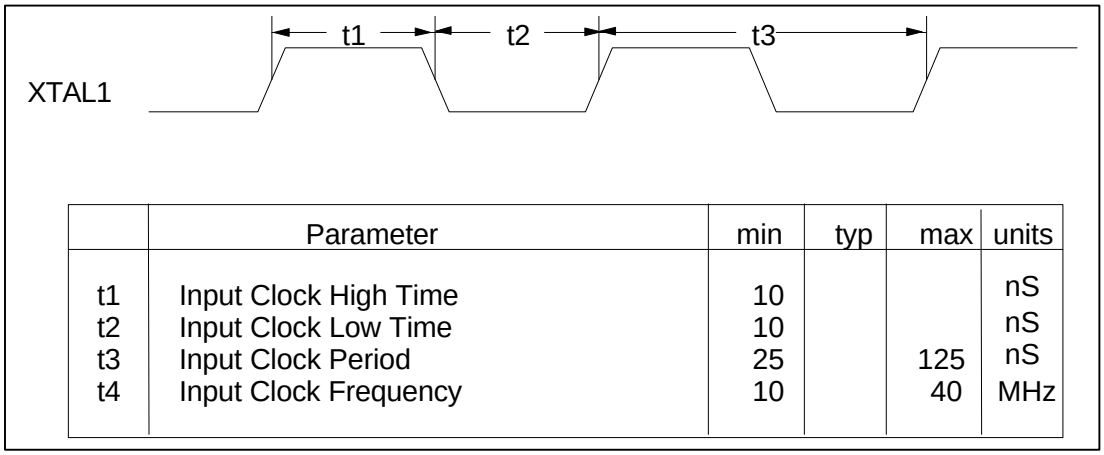


	Parameter	min	typ	max	units
t1	nPULSE2 High to TXEN Low	0		50	nS
t2	nPULSE1 Pulse Width		200*		nS
t3	nPULSE1 Period		400*		nS
t4	nPULSE2 Low to nPULSE1 Low	0		50	nS
t5	nPULSE2 High Time		100*		nS
t6	nPULSE2 Low Time		100*		nS
t7	nPULSE2 Period		200*		nS
t8	nPULSE2 High to TXEN High	0		50	nS
t9	(first rising edge on nPULSE2 after Last Bit Time) TXEN Low to First nPULSE1 Low**	650		750	nS
t10	RXIN Pulse Width	10	200*		nS
t11	RXIN Period		400*		nS

* t5,t6 = 2 x (crystal period) for clock frequencies other than 20 MHz.
* t2,t7,t10 = 4 x (crystal period) for clock frequencies other than 20 MHz.
* t3,t11 = 8 x (crystal period) for clock frequencies other than 20 MHz.
** t9: For clock frequencies other than 20 MHz, t9 = 14 x (clock period) ± 50 nsec. This period applies to data of two consecutive one's.

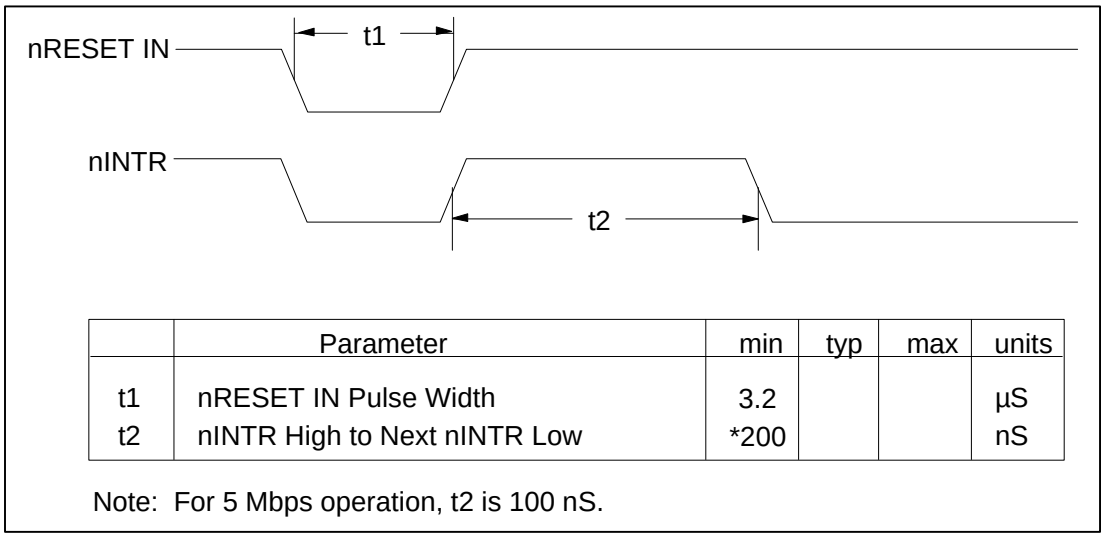
Note: Clock frequency for 5 Mbps is 40 MHz.

FIGURE 28 – BACKPLANE MODE TRANSMIT OR RECEIVE TIMING
(These signals are to and from the differential driver or the cable)



	Parameter	min	typ	max	units
t1	Input Clock High Time	10			nS
t2	Input Clock Low Time	10			nS
t3	Input Clock Period	25		125	nS
t4	Input Clock Frequency	10		40	MHz

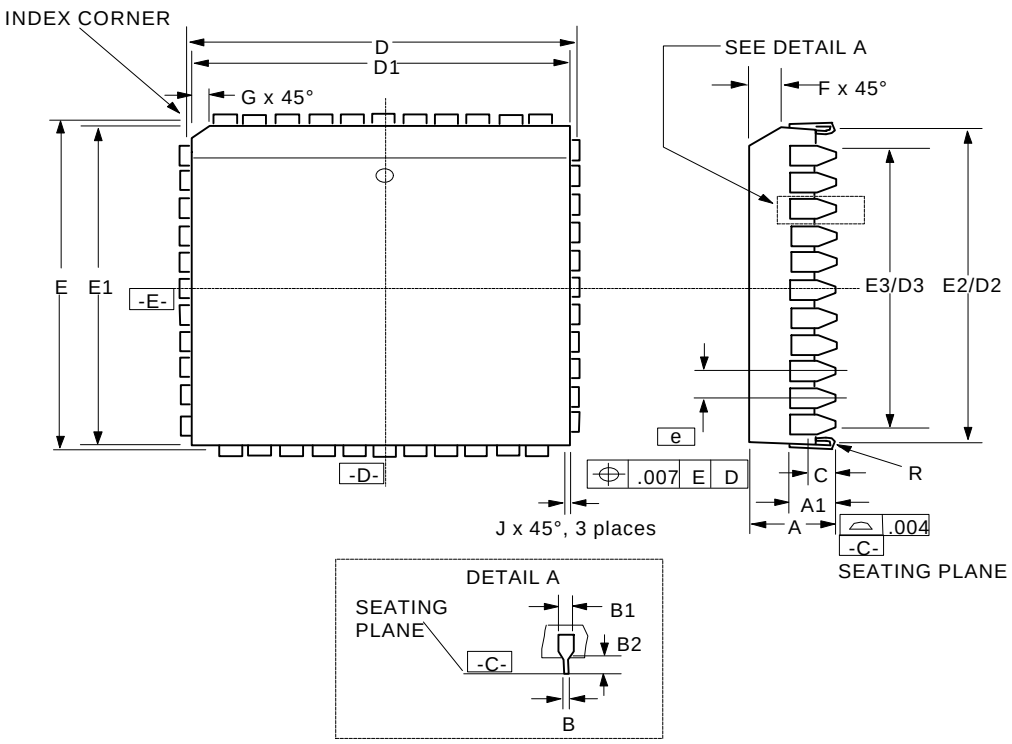
FIGURE 29 – TTL INPUT TIMING ON XTAL1 PIN



	Parameter	min	typ	max	units
t1	nRESET IN Pulse Width	3.2			µS
t2	nINTR High to Next nINTR Low	*200			nS

Note: For 5 Mbps operation, t2 is 100 nS.

FIGURE 30 – RESET AND INTERRUPT TIMING



DIM	44 LEAD
A	.160-.188
A1	.090-.120
B	.013-.021
B1	.026-.032
B2	.025 MIN
C	.020-.045
D/E	.685-.695
D1/E1	.650-.656
D2/E2	.600-.630
D3/E3	.500 REF
e	.050 BSC
F	.042-.060
G	.042-.048
J	.000-.028
R	.025-.045

Notes:
1. Dimensions are in inches.
2. Coplanarity is .004" maximum.
3. Tolerance on the position of the leads is .007" maximum.
4. Package body dimensions D1 and E1 do not include the mold protrusion. Maximum mold protrusion is .010".

FIGURE 31 – 44 PIN PLCC PACKAGE DIMENSIONS

COM20051 ERRATA SHEET

PAGE	SECTION/FIGURE/ENTRY	CORRECTION	DATE REVISED
1	Features/General Description	See Italicized Text	3/12/96
4	Overview	peer/peer changed to peer-to-peer	3/12/96
5	Description of Pin Functions/Pin No. 34 & 35	See Italicized Text	3/12/96
7	Basic Architecture	See Italicized Text	3/12/96
8	Address Decoding	See Italicized Text	3/12/96
9	Address Decode Register	See Italicized Text	3/12/96
10	ARCNET Network Core	See Italicized Text	3/12/96
11	Figure 3/See Note	See Italicized Text	3/12/96
12 & 14	Network Protocol/Network Reconfiguration	See Italicized Text	3/12/96
13	Figure 4	See Italicized Text	3/12/96
14	All Sections	See Italicized Text	3/12/96
15	All Sections	See Italicized Text	3/12/96
16	All Sections	See Italicized Text	3/12/96
17	Figure 5	See Italicized Text	3/12/96
18	System Description/All Sections	See Italicized Text	3/12/96
21	All Sections	See Italicized Text	3/12/96
23	Figure 9	See Italicized Text	3/12/96
24	Arcnet Core Functional Description/All Sections	See Italicized Text	3/12/96
25	Table 3/New Table	See Italicized Text	3/12/96
28	All Sections	See Italicized Text	3/12/96
29	All Sections	See Italicized Text	3/12/96
30	Table 7	See Italicized Text	3/12/96
34	Table 12	See Italicized Text	3/12/96
36	Table 13	See Italicized Text	3/12/96
39	Transmit Sequence	See Italicized Text	3/12/96
41	Transmit Sequence	See Italicized Text	3/12/96
42	Command Chaining	See Italicized Text	3/12/96
45	Internal Reset Logic	See Italicized Text	3/12/96
47	Last Paragraph	See Italicized Text	3/12/96
66	Generating Network Maps	See Italicized Text	3/12/96

80 Arkay Drive
Hauppauge, NY 11788
(631) 435-6000
FAX (631) 273-3123

COM20051 Rev. 03/24/2000