

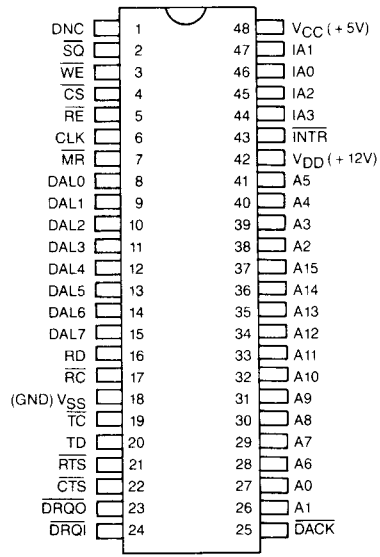
07/902

## WD2840 Local Network Token Access Controller

WD2840

### FEATURES

- Broadcast Medium Independent (Coax, RF, CATV, IR, etc.)
- Up to 254 nodes
- Dual DMA/Highly efficient Memory Block Chaining
- Token based protocol
- Acknowledge option on each datagram
- Adjustable fairness, stations may be prioritized
- Frame format similar to industry standard HDLC
- Supports Global Addressing
- Diagnostic Support: Self-Tests, System and Network
- TTL Compatible



PIN DESIGNATION

### DESCRIPTION

The WD2840 is a MOS/LSI device intended for local network applications, where reliable data communications over a shared medium is required. The device uses a buffer chaining scheme to allow efficient memory utilization. This scheme minimizes the host CPU time requirements for handling packets of data. The WD2840 frees the host CPU from extensive overhead by performing network initialization, addressing, coordination, data transmission, acknowledgements and diagnostics.

### APPLICATIONS

The WD2840 is a general purpose Local Network Token Controller applicable to virtually all types of

multi-point communications applications. The token protocol allows the sharing of one bus by up to 254 nodes. WD2840's will be designed into process control equipment, micro-computers, mini-computers, personal computers, proprietary micro-processor based applications, intelligent terminals, front-end processors, and similar equipment.

The great advantage for the design engineer is the ease with which he can implement a local network function. The WD2840 handles autonomously all major communications tasks as they relate to the local network function.

## 1.1 PIN DEFINITIONS

PIN NUMBER	SYMBOL	PIN NAME	FUNCTION
1	DNC	DO NOT CONNECT	Leave pin open.
2	$\overline{\text{SQ}}$	SIGNAL QUALITY	An active low input which signals the WD2840 that a frame may be received. The modem may negate this signal if its receive signal quality is below a reliability threshold, ensuring that the WD2840 will not accept the frame.
3	$\overline{\text{WE}}$	WRITE ENABLE	The data on the DAL are written into the selected register when $\overline{\text{CS}}$ and $\overline{\text{WE}}$ are low. $\overline{\text{RE}}$ and $\overline{\text{WE}}$ must not be low at the same time.
4	$\overline{\text{CS}}$	CHIP SELECT	Active low chip select for CPU control of I/O registers.
5	$\overline{\text{RE}}$	READ ENABLE	The content of the selected register is placed on DAL when $\overline{\text{CS}}$ and $\overline{\text{RE}}$ are low.
6	CLK	CLOCK	Clock input used for internal timing.
7	$\overline{\text{MR}}$	MASTER RESET	Initialize on active low. All registers reset to zero, except control bit ISOL is set to 1. DACK must be stable high before MR goes high. Status Register 0 is not defined at power-up (this register will be set-up upon entry into the Network mode).
8-15	DAL0-7	DATA ACCESS LINES	An 8-bit bi-directional three-state bus for CPU and DMA controlled data transfers.
16	RD	RECEIVE DATA	Receive serial data input.
17	$\overline{\text{RC}}$	RECEIVE CLOCK	This is a 1X clock input, and RD is sampled on the rising edge of $\overline{\text{RC}}$ .
18	VSS	GROUND	Ground.
19	$\overline{\text{TC}}$	TRANSMIT CLOCK	A 1X clock input. TD changes on the falling edge of $\overline{\text{TC}}$ .
20	TD	TRANSMIT DATA	Transmitted serial data output.
21	$\overline{\text{RTS}}$	REQUEST-TO-SEND	An open collector output which goes low when the WD2840 is ready to transmit either data or flags.
22	$\overline{\text{CTS}}$	CLEAR-TO-SEND	An active low input which signals the WD2840 that transmission may begin.
23	$\overline{\text{DRQO}}$	DMA REQUEST OUT	An active low output signal to initiate CPU bus request so that the WD2840 can output onto the bus.
24	$\overline{\text{DRQI}}$	DMA REQUEST IN	An active low output signal to initiate CPU bus requests so that data may be input to the WD2840.
25	DACK	DMA ACKNOWLEDGE	An active low input from the CPU in response to $\overline{\text{DRQO}}$ or $\overline{\text{DRQI}}$ . DACK must not be low if $\overline{\text{CS}}$ and $\overline{\text{RE}}$ are low or if $\overline{\text{CS}}$ and $\overline{\text{WE}}$ are low.
26-41	A0-A15*	ADDRESS LINES OUT	Sixteen address outputs from the WD2840 for DMA operation.
42	VDD	POWER SUPPLY	+ 12VDC power supply input.
43	$\overline{\text{INTR}}$	INTERRUPT REQUEST	An active low interrupt service request output. Returns high when Interrupt Register is read.
44-47	IA0-IA3*	ADDRESS LINES IN	Four address inputs to the WD2840 for CPU controlled read/write operations with registers in the WD2840. If ADRV = 0, these may be tied to A0-A3.
48	VCC	POWER SUPPLY	+ 5VDC power supply input.

**WD2840 LOCAL NETWORK  
TOKEN ACCESS CONTROLLER**

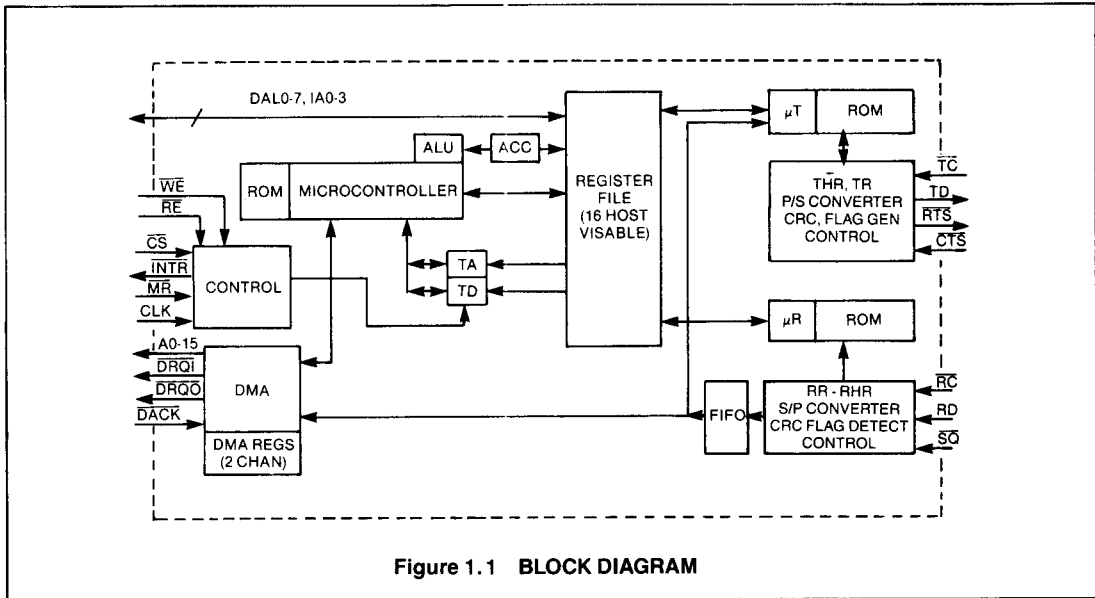
**INTRODUCTION**

The WD2840 is a single LSI device which gives systems designers the ability to include networking capabilities into their unique products simply and economically.

A general and fundamental advantage to the use of complex LSI in a given system is the partitioning of required technical expertise. A successful user of the WD2840 need not be a data-communications expert, and further, he need not be at all concerned with low level network details (though these details are documented and available to him if he is interested). The potential user of the WD2840 must simply evaluate the communications facilities provided by the device to determine its suitability for the intended use.

The WD2840 is designed to logically interconnect 2 to 254 user devices over a shared communications medium. Examples of typical mediums include coax cable, twisted pair bus, RF, and CATV. All network control functions, such as data framing and error checking, destination filtering, fair and adjustable transmission scheduling, and network initialization and fault recovery (caused by noise for example) are handled completely by the WD2840.

The protocol implemented allows guaranteed station access intervals allowing applications in factory automation and other critical communications environments where "statistical delays" are not acceptable. The WD2840 token protocol also allows the addition and/or removal of stations to a network at anytime, including while operating.



**Figure 1.1 BLOCK DIAGRAM**

Serious attention has also been given to the user's interface to the device. The interface is a combination of conventional I/O registers and an elaborate DMA buffer chaining interface. This chaining feature allows the user much more efficient use of his system memory, particularly in situations where the maximum message sent over the network is much longer than the average size. This feature also allows the automatic queuing of messages independently of the user's consumption rate, in effect, speed decoupling the user's CPU and processing requirements from the network.

The WD2840 has several parameters (registers) that allow tailoring to the user's requirements. In this way, network priority and access ordering, to name two, can be manually set if desired.

Using an integrated version of these network algorithms saves not only the development costs already mentioned, but further, the total processing power required for the user's application is not increased. In other words, a CPU upgrade can likely be avoided by "distributing" the network processing task into LSI devices such as the WD2840.

### SCOPE

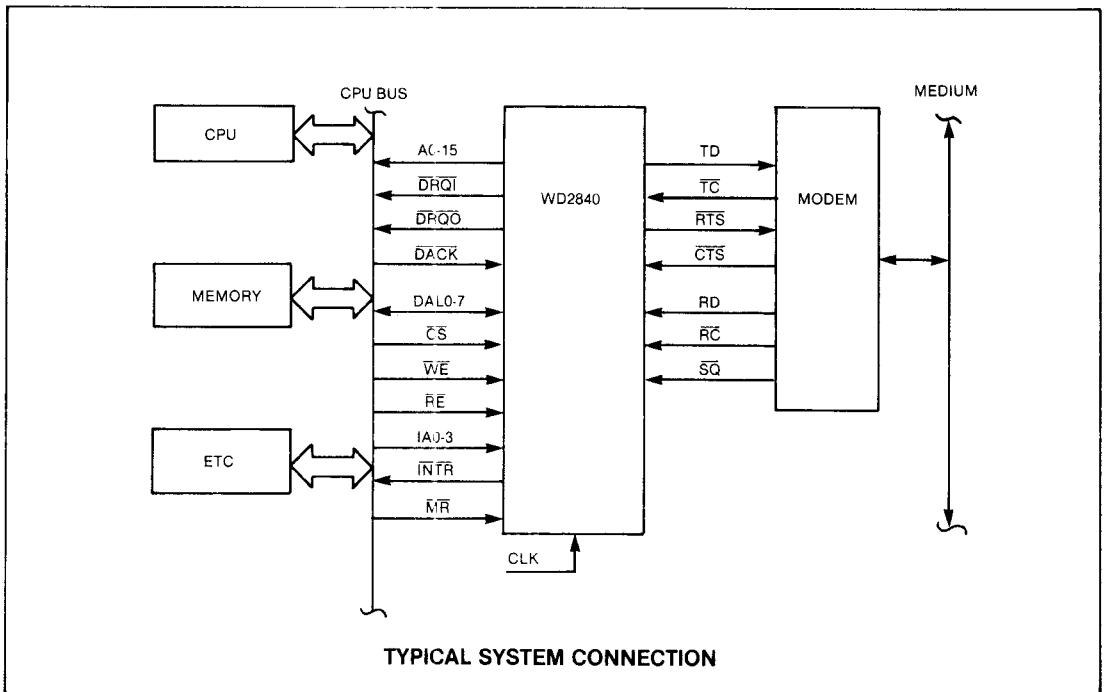
This document differs from traditional LSI data sheets in that it details not only the LSI implementation of a function, but also defines the overall function in detail. Specifically, this document includes de-

tails of the communications protocol implemented by the WD2840 Token Access Controller.

The document is organized into three main sections: SECTION ONE is much like a traditional data sheet including register descriptions, pin definitions, and hardware architecture.

SECTION TWO describes the interfaces to the WD2840. The network side is conventional, the host side consists of an elaborate DMA interface with control blocks and WD2840/host handshaking.

SECTION THREE details the network protocol implemented by the device. Normal operation, initialization, and the handling of error conditions are described.



## 1.2 DEVICE ARCHITECTURE

A detailed block diagram of the WD2840 is shown in Figure 1.1.

Mode control and monitor of status by the user's CPU is performed through the Read/Write Control circuit, which reads from or writes into registers addressed by IA0-IA3.

Transmit and receive data are accessed through DMA control. Serial data is generated and received by the bit-oriented controllers.

Internal control of the WD2840 is by means of three internal micro-controllers; one for transmit, one for receive, and one for overall control.

Parallel transmit data is entered into the Transmitter Holding Register (THR), and then presented to the Transmitter Register (TR) which converts the data to a serial bit stream. The Frame Check Sequence (FCS) is computed in the sixteen bit CRC register, and the results become the transmitted FCS.

Parallel receive data enters the Receiver Holding Register (RHR) from the 24 bit serial Receive Register (RR). The 24-bit length of RR prevents received FCS data from entering the RHR. The receiver CRC register is used to test the validity of the received FCS. A three level FIFO is included in the receiver.

The WD2840 sends all information, network control and user data, in blocks called frames. Each frame starts and ends with a single flag (binary pattern 01111110). In between flags, data transparency is provided by the insertion of a zero bit after all sequences of five contiguous one bits. The receiver will strip the inserted zero bits. (See section on frame format for location of address, control, and FCS fields.)

## 1.3 REGISTER DEFINITION

The WD2840 is controlled and monitored by sixteen 8 bit registers. This set of registers consists of two Control Registers, three Status Registers, an In-

terrupt Event Register, a Counter Register and a variety of Parameter Registers. In general the host is responsible for defining these registers (except certain host read-only registers: SR0-2, IR0, CTR0 and NA) to contain proper and meaningful values prior to entering Network Mode from Isolate State. Furthermore, while the WD2840 is in Network Mode, the CBP (H,L) and MA registers must not be changed by the host. Register NAR may be changed arbitrarily but will only be considered by the WD2840 in response to the NEWNA (CR10) control bit being set. The two Control Registers and the TA, TD, AHOLT, TXLT registers may change dynamically to control the behavior of the WD2840.

REG [1]	NAME	DESCRIPTION
0	CR0	Control Register 0
1	CR1	Control Register 1
2[2]	SR0	Status Register 0
3[2]	IR0	Interrupt Event Register
4[2]	SR1	Status Register 1
5[2]	SR2	Status Register 2
6[2]	CTR0	Counter Register 0
7[2]	NA	Next Address
8	TA	ACK Timer
9	TD	Net Dead Timer
A	CBPH	Control Block Pointer (MSB)
B	CBPL	Control Block Pointer (LSB)
C	NAR	Next Address, Request
D	AHOLT	Access Hold-off Limit
E	TXLT	Transmit Limit
F	MA	My Address

[1] = Hexadecimal representation of IA0-IA3.

[2] = CPU read only, write not possible.

Control, status, and interrupt bits will be referred to as CR, SR, or IR, respectively, along with two digits. For example, SR21 refers to status register #2 and bit 1, which is "STATE."

## SUMMARY — CONTROL, STATUS, INTERRUPT REGISTERS

REGISTER	7	6	5	BIT # 4	3	2	1	0
CR0	TXDEN	TXEN	RXEN	ITOKON	ILOOP	COPY	NOINT	ISOL [1]
CR1[2] CR1[4]	DIAGC DIAGC	PIGT 0	INIT 0	ADRV ADRV	GIRING DMAT	0 LOOP	TOFF RAMT	NEWNA NUDIAG
SR0	LASTF	SENDACK	L2	0	BSZ3 ...	BSZ2 ...	BSZ1 ...	BSZ0
IR0 [3]	ITERR	IROR	INS	ITRAN	IREC	ITOK	ITA	ITD/M
SR1	TAOUT	$\overline{\text{IRTS}}$	RECIDL	1	1	1	1	1
SR2	NXTT0	NXTR0	TR	ACKRQ	RETRY	TSENT	STATE	INRING

NOTE: ZERO BITS (0) SHOWN ABOVE ARE RESERVED AND SHOULD NOT BE USED.

## NOTES:

[1] = Set to 1 on power-up or master reset.

[2] = Non diagnostic mode only (CR17-DIAGC cleared).

[3] = Any bit set causes host interrupt ( $\overline{\text{INTR}}$  goes true) when Master Interrupt Suppress (CR01) is clear. All bits are cleared when register is read by the host.

[4] = Diagnostic State only (CR17-DIAGC set). See diagnostic section for register usage in diagnostic mode.

## CR0 — CONTROL REGISTER 0 DEFINITION

REGISTER	CR07	CR06	CR05	CR04	CR03	CR02	CR01	CR00
CR0	TXDEN	TXEN	RXEN	ITOKON	ILOOP	COPY	NOINT	ISOL

BIT	NAME	DESCRIPTION
CR00	ISOL	Isolate. Set true on power up or master reset. Host clears this bit after the host memory based WD2840 control block and other WD2840 registers have been set up. May be set by the host at any time (will be ignored if WD2840 is in diagnostic state). There is some delay for the WD2840 to respond to any state change request. A state change to network mode is acknowledged by the state confirmation status bit (SR21-STATE) being cleared. Setting ISOL while the WD2840 is in Network State will cause a state change to Isolate State, confirmed by an interrupt event (IR00-ITM) and the STATE status bit (SR21) being set. This transaction will be delayed until the node does not possess the token. Any in-progress frame transmission will be completed normally (at the current frame, regardless of queue length), followed by a normal token pass sequence.
CR01	NOINT	Master Interrupt Suppress. When clear, the WD2840 will generate host interrupt requests ( $\overline{\text{INTR}}$ low) if any bit in the WD2840 interrupt request register (IR0) is set. When set, only the interrupt request is suppressed, not the setting of bits in IR0. Note that any interrupt request will be dropped by the WD2840 when IR0 is read since this will clear IR0.
CR02	COPY	Enables COPY mode. When set causes all received data frames to be accepted and DMA'ed into memory regardless of destination address. (See description in Diagnostics Section.)
CR03	ILOOP	Instructs the WD2840 to loop data internally from transmitter to receiver. Used with the LOOP diagnostic. Must NOT be set while in network mode (CR00-ISOL clear).
CR04	ITOKON	Enable Token received interrupts. When clear no Token received interrupts are generated. When set the WD2840 generates an Itok interrupt when a token is received.

BIT	NAME	DESCRIPTION
CR05	RXEN	<p>Receive Data Enable. When clear, the WD2840 still makes normal responses to supervisory frames (scan, token pass), but will not DMA any data frames into memory and ignores the receiver buffer chain. However any data frame which is addressed to this node and for which an ACK is requested, will be NAK'ed with a "receiver not enable" Nak code. When RXEN is set, it allows the receiver to DMA appropriate data frames into memory. RXEN may be arbitrarily set and reset while in Network State but changes will not affect any frames in progress.</p> <p>NOTE: Even when RXEN is clear, the WD2840 is "following" the receiver buffer chain with an internal register pointing either to the next available buffer (NXTR0 set) or, if the chain is exhausted, to a link field of zero (NXTR0 clear). The constraints on host manipulation of the receiver buffer chain are the same regardless of the state of RXEN. See the subsequent section on Receiver Memory Interface for more details.</p>
CR06	TXEN	<p>Master Transmit Enable. When clear no transmissions will occur and the transmit buffer chain will be ignored. When set, transmission activity is further dependent upon TXDEN (CR07).</p> <p>NOTE: Even when TXEN is clear, the WD2840 is "following" the transmitter buffer chain with an internal register pointing either to the next frame to transmit (NXTT0 set) or, if the chain is exhausted, to a link field of zero (NXTT0 clear). The constraints on host manipulation of the transmitter buffer chain are the same regardless of the state of TXEN. See the subsequent section on Transmitter Memory Interface for more details.</p>
CR07	TXDEN	<p>Data Transmit Enable. Has no meaning unless TXEN is set. When set in conjunction with TXEN, normal WD2840 transmission of data and supervisory frames will occur. When clear and with TXEN set, only data frame transmission will be suppressed. That is, token pass and Ack/Nak supervisory frames will still be transmitted when appropriate.</p> <p>NOTE: The note above for TXEN applies.</p>

#### CR1 — CONTROL REGISTER 1 DEFINITION

REGISTER	CR17	CR16	CR15	CR14	CR13	CR12	CR11	CR10
CR1 CR1	DIAGC DIAGC	PIGT 0	INIT 0	ADRV ADRV	GIRING DMAT	0 LOOPT	TOFF RAMT	NEWNA NUDIAG

BIT	NAME	DESCRIPTION (CR17 = 0, Network mode)
CR10	NEWNA	<p>Update NA register. When set causes WD2840 to copy the contents of register NAR into register NA. The WD2840 clears this bit after the function is complete. This mechanism allows the host to define the WD2840's successor in the logical ring. The node's next token pass will be to the new NA node.</p> <p>NOTE: The normal token pass recovery applies. If the token pass to the new NA is not successful, a normal scan sequence will occur where the WD2840 attempts a single token pass to each node address in numerical sequence until a successful pass occurs or the node's address itself is reached.</p>
CR11	TOFF	<p>When set causes WD2840 to ignore timers. (This is NOT intended to be used in an operational network, but is provided to support network diagnosis.) CAUTION: This control bit disables all automatic network error recovery.</p>
CR12	—	(Not used, Reserved.)
CR13	GIRING	<p>Get in logical ring. Instructs the WD2840 to gain entry into the logical ring at the next opportunity (i.e. respond to a token pass). The INRING status bit (SR20) is confirmation; when INRING is set, it indicates that the WD2840 is participating in a logical ring of at least two nodes. If the host clears GIRING while INRING is set, the WD2840 will not accept the next token pass to it at which time INRING will be cleared as confirmation.</p>

BIT	NAME	DESCRIPTION (CR17 = 0, Network mode)
CR14	ADRV	Address Driver Enable. Enables the sixteen output address (A0-A15). If ADRV = 0, the outputs are tri-state and are in HI-Z, except when <b>DACK</b> goes low. If ADRV = 1, the outputs are always TTL levels.
CR15	INIT	<p>Network Initialization Enable. When clear, the WD2840 will not attempt to (re)initialize the network if the net dead timer (TD) expires. When set, TD timer expiration causes the WD2840 to enter Scan Mode. In this mode it transmits a token pass frame to each node numerically higher in address, one after another, until either network activity occurs (another node responds) or until the node's own address is reached. When Scan Mode begins, the first node address used is the then current NA (Next Address) node address. This value is derived from and is affected by the following actions:</p> <ol style="list-style-type: none"> <li>1. At transition into Network State it defaults to MA + 1.</li> <li>2. It may be set by the host using the NAR register and the NEWNA (CR10) control flag.</li> <li>3. Upon receipt of a Scan Mode frame, NA is redefined to MA + 1.</li> </ol> <p>The successful initialization of the network by Scan Mode causes NA to be defined as the first responding node (hence, this node's successor).</p> <p>All node address computations are ascending and circular within the valid node address range of 1-254.</p> <p>NOTE: Since this network initialization activity comes about because of a timer expiration, TOFF (CR11) must be clear.</p>
CR16	PIGT	If set, instructs WD2840 to piggy back token on last data frame transmitted. This request is honored if the last frame is determined as a result of limit TXLT or the LAST bit set in the TX-FCB, but not if transmission ends due to the reaching of the end of the chain.
CR17	DIAGC	Enables diagnostic mode. In network mode this bit must be zero.

#### CR1 — CONTROL REGISTER 1 DEFINITIONS

BIT	NAME	DESCRIPTION (CR17 = 1, Diagnostic mode)
CR10	NUDIAG	Perform a new diagnostic. When set causes WD2840 to perform the selected diagnostics. The host initializes the appropriate registers for the particular diagnostic and by setting this bit can initiate the test. The WD2840 clears this bit after completion of the diagnostic.
CR11	RAMT	Selects internal RAM test if in diagnostic mode.
CR12	LOOPT	Selects Loop Test if in diagnostic mode.
CR13	DMAT	Selects DMA Test if in diagnostic mode.
CR14	ADRV	Address Driver Enable. Enables the sixteen output address (A0-A15). If ADRV = 0, the outputs are tri-state and are in HI-Z, except when <b>DACK</b> goes low. If ADRV = 1, the outputs are always TTL levels.
CR15	—	(Not used, Reserved.)
CR16	—	(Not used, Reserved.)
CR17	DIAGC	Enables diagnostic mode. Confirmation of diagnostic mode is via status bit STATE (SR21). When DIAGC and STATE are both set, diagnostic functions of CR1 apply. When DIAGC is cleared, after the selected set of diagnostics in progress complete, the WD2840 will transition to the Isolate state. This transition will cause an interrupt event (ITM).

## SR0 — STATUS REGISTER 0 DEFINITION

REGISTER	SR07	SR06	SR05	SR04	SR03	SR02	SR01	SR00
SR0	LASTF	SENDACK	L2	0	BSZ3	BSZ2	BSZ1	BSZ0

BIT	NAME	DESCRIPTION
SR00	BSIZ	BSIZ0 — BSIZ3
..		
SR03		Buffer size, defines the buffer size in multiples of 64 bytes (the value ranges from 0 to 15H. Corresponding to a buffer size of 64 to 1024 bytes in 64 byte increments). This value is used internally to define buffer boundaries to allow the chip to link buffers. A maximum of 16 buffers may be used for a single frame.
SR04	—	Not used.
SR05	L2	An internal flag set during frame transmission if the length value of the current frame is equal to eight. For normal data frame transmission this means the frame has no data field and for transparent frame transmission this means the frame is an access control frame. (SCAN FRAME)
SR06	SENDACK	An internal flag set during data frame reception to indicate that the incoming frame should be acknowledged (send ack/nak frame). This flag is cleared when the acknowledgement has been transmitted.
SR07	LASTF	An internal flag set during data frame transmission to indicate that the current frame will be the last to be transmitted with this token. Five situations can cause this to occur: 1) ISOL (CR00) becoming set, 2) TXDEN (CR07) becomes clear, 3) current frame flagged (via FCB) to be "last frame," 4) the current token frame count reaching the TXLT limit, 5) transmitter under-run detection. Note in particular that the last frame in the transmit queue will not cause LASTF to set since it's being last is not known until frame end. Also if a piggy-back token is permitted (CR16 set) and no acknowledgement is requested (via FCXB), the token will be piggybacked on the current (last) data frame. LASTF is not cleared until the next data frame transmission begins.

## IR0 — INTERRUPT REGISTER DEFINITION

REGISTER	IR07	IR06	IR05	IR04	IR03	IR02	IR01	IR00
IR0	ITERR	IROR	INS	ITRAN	IREC	ITOK	ITA	ITD/M

The setting of any bit in this register by the WD2840 causes an interrupt request ( $\overline{\text{INTR}} = \text{low}$ ) if NOINT (CR01) is clear. The reading of this register by the host clears all bits (and any interrupt request).

BIT	NAME	DESCRIPTION (1)
IR00	ITD/M	Network dead or mode change (dual use). When in Network mode, timer TD expiring (with TOFF clear) causes this bit to be set to indicate no network activity has occurred within the timeout period. Also INRING (SR20) is cleared and, if INIT (CR15) is set, the WD2840 will enter Scan Mode (see INIT - CR15 for details). Transition from Network or Diagnostic State to the Isolate State will be confirmed by this interrupt. The choice between the ITD and ITM interpretations is easily made based on the ISOL (CR00) bit.
IR01	ITA	Date Frame Transmission Unsuccessful. This interrupt indicates that a transmitted data frame with an acknowledge request was not successfully acknowledged. Either a NAK or no response after two transmissions will cause this. The exact cause can be determined by inspecting the appropriate FSB.
IR02	ITOK	The token has been received.

BIT	NAME	DESCRIPTION (1)
IR03	IREC	Data Frame Received. This interrupt signifies that a good data frame has been properly received and DMA'ed into the buffer chain. Frames that have been received can be identified by following the buffer chain noting the WD2840 frame status bytes (FSB). A non-zero FSB (host must clear when queuing free buffers) indicates a properly received frame. The host may freely remove all received frames from the chain up to but NOT necessarily including the last one posted. The last one posted may only be removed if the WD2840 NXTR0 (SR26) is set. For more details see the explanation for NXTR0.
IR04	ITRAN	Indicates that at least one data frame has been transmitted. The number of frames transmitted and the status of each (i.e. ACK/NAK, retry count) is determined by following the transmit chain and inspecting frame status bytes (FSB). All transmitted frames up to but NOT including the last posted may be freely removed. The last one posted may only be removed if the WD2840 NXTT0 (SR27) is set. For more details see the explanation for NXTT0.
IR05	INS	New successor. The WD2840 has identified a new successor in the logical ring. This happens when the prior successor either failed to respond to a token pass or as instigated by a network scan frame.
IR06	IROR	Receiver over-run. The WD2840 ran out of buffers or access to the DMA channel was delayed by the host so long as to cause loss of received data.
IR07	ITERR	Transmitter error. Three abnormal frame transmission cases can cause the ITERR interrupt. The causes are "transmitter underrun," "premature end of chain," and "exceeded 16 buffers." The frame transmission will repeat once per token until the host removes the WD2840 from the network, or the cause of the error is fixed.

(1) = Non diagnostic mode only. See diagnostic section for register usage for diagnostics.

#### SR1 — STATUS REGISTER 1 DEFINITION

REGISTER	SR17	SR16	SR15	SR14	SR13	SR12	SR11	SR00
SR1	TAOUT	$\overline{\text{IRTS}}$	RECIDL	1	1	1	1	1

BIT	NAME	DESCRIPTION
SR10 :: SR14	—	(Not used, reserved.)
SR15	RECIDL	Receiver Idle. Indicates the WD2840 has received at least 15 contiguous ones.
SR16	$\overline{\text{IRTS}}$	Internal Request To Send. Indicates the transmitter is attempting (successful or not) to send either data or flags. If the RTS pin is not tied to ground or WIRE-OR'ED with another signal, then $\overline{\text{IRTS}} = \text{RTS}$ .
SR17	TAOUT	Timer TA expired.

## SR2 — STATUS REGISTER 2 DEFINITION

REGISTER	SR27	SR26	SR25	SR24	SR23	SR22	SR21	SR20
SR2	NXTTO	NXTRO	TR	ACKRQ	RETRY	TSENT	STATE	INRING

BIT	NAME	DESCRIPTION
SR20	INRING	In logical ring. Indicates the node has had the token and has successfully passed it at least once (therefore it is included in a logical ring of at least two nodes). See GIRING (CR13) for other comments.
SR21	STATE	Mode confirmation. Depending on DIAGC (CR17), the WD2840 is either in Isolate or Diagnostic state. When ISOL (CR00) is set, STATE set confirms the WD2840 is not in Network State. When ISOL is clear, STATE clear confirms Network State. Note any state transition into Isolate State causes an interrupt event to occur (ITM).
SR22	TSENT	An internal flag. TSENT is set when the WD2840 passes the token. It may have been either a piggyback or explicit token pass frame. TSENT is cleared when the next frame is received.
SR23	RETRY	An internal flag which is set when either a data frame or a token pass frame must be retransmitted. Data frames are only retransmitted if they have an acknowledge request and no response at all occurred. Token pass frames (except Scan) are retransmitted if no network activity was detected. Both of these situations are detected as a result of a TA timeout.
SR24	ACKRQ	An internal flag set during data frame transmission if an acknowledgement is requested for the specific frame. If this is the case, the WD2840 pauses to await the ACK/NAK response frame; if the TA timer expires before the response, a single retry will occur (see RETRY-SR23). ACKRQ is not cleared until the beginning of the next data frame transmission.
SR25	TR	An internal flag set when the WD2840 receives a token passed to it. It is cleared when the token is passed (or if it is ignored for any reason. For example, piggyback token on a bad data frame, TXEN clear, or detection of duplicate tokens in the logical ring).
SR26	NXTRO	Internal Receive Buffer Pointer State. Because of the linked list approach used in the buffer chains, the WD2840 internal register used to follow the list is either pointing to the next buffer in the chain or at the address of the next buffer in the chain (prior buffer's link field). The WD2840 will always advance along the chain so that it has the address of the next buffer to be used. However, when a zero link is encountered, the WD2840 retains the link field address expecting eventually that the chain will be extended by the host making the link some non-zero value. When the WD2840 actually needs the next buffer, it looks again at the contents of the link field expecting it to have been changed (chain extended) to the address of an available buffer. The NXTRO bit differentiates between these two situations. When set it indicates the WD2840 has the address of the next buffer and that all prior frames (denoted by posted FSB's) can be removed from the chain for received frame processing by the host. When NXTRO is clear it indicates that the WD2840 has advanced to a zero link (end of chain).  NOTE: In this situation, the last posted frame CANNOT be removed from the chain for processing since it is the link field of his last buffer that must be set in order to extend the receiver buffer chain.
SR27	NXTTO	Internal Transmit Buffer Pointer State. The comments for NXTRO (SR26) apply (in an analogous manner) to NXTTO since the transmit buffer chain is handled by the WD2840 using an identical scheme. When NXTTO is set it indicates that the WD2840 has the address of the next frame to transmit in its internal register. However when clear, it indicates that the transmit chain internal register points to the link field of the last buffer of the last transmitted frame. This link field contained zero when first read. For the transmit case, this is a normal situation corresponding to no data frames to transmit.  NOTE: As in the receive case, when NXTTO is set, all previously transmitted frames (denoted by posted FSB's) can be removed from the chain for reuse. However, when NXTTO is clear it indicates that the transmit chain must be extended by the host before removing the very last frame that has been transmitted (posted).

## OTHER REGISTER DEFINITIONS

NAME	DESCRIPTION
CTR0	Running Limit Counter. Used by the WD2840 for Access Hold-Off Limit (AHOLT) checking and Transmit Limit (TXLT) checking. When transmitting data frames CTR0 is used for TXLT counting; otherwise it is used for AHOLT counting. The counter runs from zero to the 8-bit limit value.
NA	Next Address. This register shows the current (instantaneous) successor node in the network logical ring. For validity, the WD2840 should be "in the ring" (see GIRING - CR13 and INRING - SR20 for more details). The successor node may be changed for a variety of reasons: <ol style="list-style-type: none"> <li>1. Any attempted token pass that fails twice will cause the WD2840 to attempt to locate a new successor by sequentially trying token passes to successively higher node addresses beginning with NA + 1.</li> <li>2. A received Scan frame will cause NA to be set to MA<sup>n</sup> + 1. If the next token pass fails case 1 applies.</li> <li>3. The host may arbitrarily redefine NA by using the NAR register and the NEWNA (CR10) control bit. At a convenient point the WD2840 recognizes NEWNA, copies NAR into NA, then clears NEWNA as confirmation. If the next token pass fails case 1 applies.</li> </ol>
TA	Acknowledgement Timer. Value of maximum allowed time between frame transmission and ACK/NAK (if requested), or between token sent and network activity. The delay is in increments of 64 times the period of the clock CLK. Thus, if CLK = 2 MHz, then TA may be set in increments of 32 microseconds (range of 32 $\mu$ s to 8.2 ms).
TD	Network Dead Timer. Value of maximum time interval between received valid frames on the network. 32X range of TA.
CBP (H,L)	Control Block Pointer. A sixteen bit pointer to the WD2840 control block in the user's memory. Must not be modified while the WD2840 is in network mode.
NAR	Next Address, Request. Used in conjunction with the NEWNA (CR10) control bit to cause the WD2840 to update the NA register. This redefines the node's successor in the network logical ring. It MUST be an address in the range 1-254. The acceptance of this update is confirmed when the NEWNA control bit is cleared. On the next token pass, if the redefined successor fails to accept the token, this WD2840 enters Scan mode where it sequentially attempts a token pass to successively higher nodes.
AHOLT	Access Hold-off Limit. This register is set at a value indicating the number of access cycles (tokens received) that must be skipped before the data frame may be transmitted. (A token pass frame will be sent even if a data frame may not be sent at a given access cycle.) Initialized to zero at power up.
TXLT	Transmit Limit. This register is set at the maximum number of consecutive data frames the WD2840 may transmit during one access cycle. A value of zero allows the WD2840 to transmit all frames queued up to 256. Initialized to zero at power up.
MA	My Address. The WD2840 receives only frames with this destination address (along with the broadcast address) and inserts this address into the SA field of any transmitted frame. Must be set by the host (range is 1 to 254).

### 1.4 DIAGNOSTIC AIDS

There are three levels of diagnostics supported by the WD2840; those that are associated with the network as a whole, those associated with the in-

DIAGNOSTIC MODE CONTROL			DEFINITION
CR00 ISOL	CR17 DIAGC	SR21 STATE	
1	0	0	WD2840 "Isolated." Power-up condition or isolate request.
0	0	0	WD2840 active.
1	0	1	Isolate request function confirmed.
1	1	0	Host request to enter diagnostic mode.
1	1	1	Diagnostic mode confirmed. Diagnostic functions of CR1 apply.
0	0	1	Illegal.
0	1	0	Illegal.
0	1	1	Illegal.

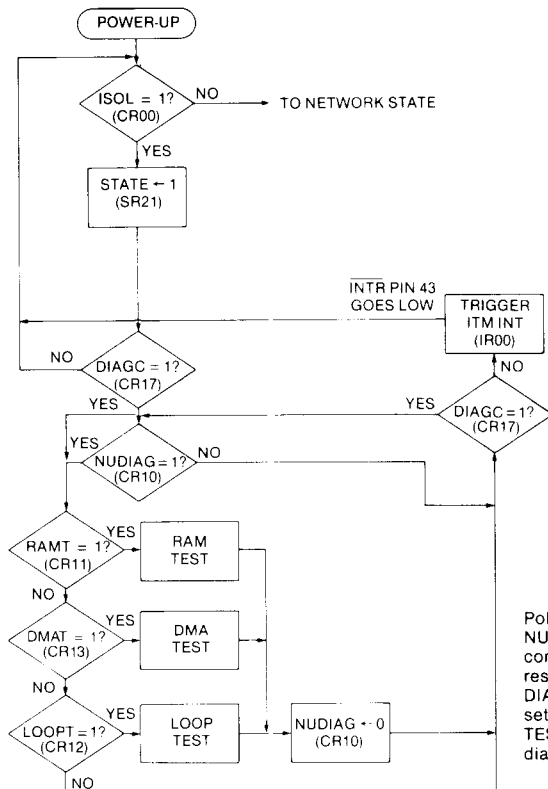
dividual node, and those that are limited to the WD2840 as a device. These tests are Network Diagnostics, System Diagnostics and Self Diagnostics respectively. The Network Diagnostics can be performed while the WD2840 is in the logical ring, but the System Diagnostics and the Self Diagnostics may be used only while the WD2840 is in the diagnostic mode.

Diagnostic mode may be entered after power-up or from the network mode by manipulation of the mode control bits. The mode transition is confirmed by the WD2840 via the STATE status bit.

Once in diagnostic mode, the desired test is selected via CR1. Because most of registers 8 through F are interpreted differently for each test, only one of the diagnostic test bits should be set at a time. In conjunction with setting the diagnostic bits, the NUDIAG (CR10) bit must be set to perform the diagnostic test requested.

At the completion of the selected test NUDIAG is cleared by the WD2840. Therefore the host can initiate a diagnostic by entering the diagnostic mode, initializing the proper registers, setting the desired diagnostic bit, and setting NUDIAG. The host then monitors CR1 for NUDIAG going to zero, indicating the completion of the requested diagnostic.

**DIAGNOSTIC STATE FLOW CHART**



Poll CR1 for 0 in NUDIAG for test complete, process results, then clear DIAGC in CR1 to exit or set NUDIAG and NEXT TEST to continue diagnostics.

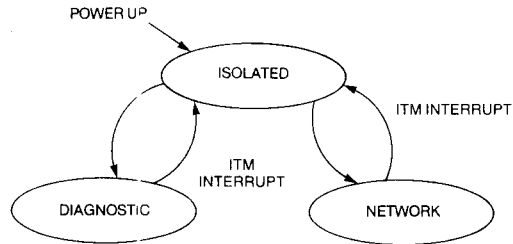


Figure 1.3 FUNCTIONAL STATES

### 1.4.1 SELF DIAGNOSTICS

#### Internal Ram and Interrupt Test

There are nine eight bit registers in the WD2840 which are not directly accessible by the users CPU. This test provides a means to check those registers and the interrupt register. The contents of register A are placed into the interrupt register and five even internal registers, and the contents of register B in four odd internal registers. The nine registers are then added together without carry and the result is placed in registers 2, 5, 6, 7.

Use the following procedure to initiate the RAM test:

1. Enter diagnostic mode.
2. Set up registers A and B
3. Set RAMT.
4. Set NUDIAG (can be set with RAMT bit together).
5. Wait for NUDIAG to be cleared.
6. Read registers 2, 5, 6, 7. Clear RAMT.

Note that the setting of any bit in the interrupt register while **NOINT** is clear will generate a hardware interrupt (INTR, pin 43 goes true).

### 1.4.2 SYSTEM DIAGNOSTICS

#### DMA Test

This test verifies proper operation of the DMA sub-system by reading the value from a register and writing it into the user memory. The test continues by reading the value from the same location in memory and writing it into another register.

The value is read from register C. Using the transmitter DMA sub-system, it is written into memory location addressed by register A and B (location N; register A is the MSB). The receiver DMA sub-system is used and contents of the same address is read and it is stored into the register 7. Next the receiver dma is used and the contents from register D is written into location N + 1. The transmitter dma reads the value from location N + 1 and stores it into register 6.

It is the host's responsibility to check if the contents of registers C and register 7 and memory location N

match. The same is true for registers D and 6 and memory location N + 1.

#### Loop-Back Test

The host can test the WD2840 transmitter and receiver logic by using the Loop Test.

There are two Loop Tests available for diagnostic purposes — internal and external.

(CR12) LOOP	(CR03) 1LOOP	DEFINITION
0	0	Not in Loop Test
0	1	Do not use in network mode
1	0	External loop
1	1	Internal loop

When using the external loop the interface or modem must have the necessary logic to tie TD to RD and TC to RC.

Use the following procedure to run the loop test.

1. Set up a 256 byte transmit buffer with the data pattern to be transmitted.
2. Initialize a 256 byte receive buffer with all "00s" or "FFs."
3. Load register A (MSB) and B (LSB) with the address of the transmit buffer.
4. Load register C (MSB) and D (LSB) with the address of the receive buffer.
5. Load register 0 for Internal or External Loop.
6. Load register 1 for diagnostic & loop. (85H)
7. Refer to Diagnostic State Flow Chart.

#### NOTE:

If this test frame is allowed onto the network, transmission collisions may occur. Further, the first three bytes of the transmit buffers will be interpreted as TC, DA and SA, respectively, by the other stations. Therefore in case this test is initiated while this node is in the logical ring, care should be taken for choosing these three values for external loop-back test.

For proper operation of the internal loop-back test the CTS and SQ pins of the WD2840 should be either tied to ground or tied to RTS pin of the WD2840.

### 1.4.3 NETWORK DIAGNOSTICS

#### Duplicate Station Detection

Duplicate stations (more than one station with the same address) can result from the faulty programming of internal register MA (due to wrong address switch settings on the user's device, for example). This is expected to occur often enough to warrant the addition of a detection algorithm in the users WD2840 initialization procedure.

After initialization, the user should place the WD2840 in the network mode with TXEN off and ITOKON on. This will cause the WD2840 to generate an ITOK interrupt each time a token is passed to its address (MA). The host must provide the timeout algorithm which should be greater than the maximum time for the network to pass the token around the ring twice. Checking twice eliminates the possibility that the Network is in the scan mode and sending tokens to non-existing stations.

It is useful to note that this constraint requiring each node which is participating in the network logical ring to have a unique address does not extend to nodes which are "listening" but not "in the ring." It might be useful to a network designer to have groups of receive only nodes which have the same node address but do not participate in the network token passing (see GIRING - CR13). Data frames transmitted to such clusters must not request acknowledgement since all nodes in the cluster would simultaneously respond.

#### Copy Mode

The COPY Mode is selected by setting the COPY control bit (CR02). Normally the WD2840 receives (DMA's into the receive buffer chain) data frames only if they contain the general broadcast destination address or if they are specifically addressed to the WD2840. This occurs when the frame's destination address (DA) matches the WD2840 my address (MA, set by the host).

However, when COPY mode is selected data frames which are specifically addressed to other nodes will be treated as broadcast frames by this node. The COPY mode allows a specific node to "eavesdrop" on data frame traffic on the network.

#### Nak Response

The WD2840 sends negative acknowledgements (NAK's) on response to received frames under

several circumstances. The NAK prevents the transmitting node from wasting bandwidth retrying indiscriminately, and further, lends visibility to individual network node problems. The NAK includes a reason code which is available to the transmitter's software (via the TFSB).

Each data frame to be transmitted can be specifically marked (via the FCB) by the host to require an ACK/NAK response from the receiving WD2840. In the absence of errors, an acknowledge (ACK) frame will be returned to the transmitter as confirmation. However, several circumstances cause a Negative Acknowledge (NAK) to be returned:

1. Insufficient buffer space
2. Receiver not enabled (RXEN - CR05 cleared)
3. Receiver overrun
4. Frame exceeded 16 buffers in length

This information is placed in the transmitted frames's FSB. See section 2.1.2 for more details on the Transmit Frame Status Byte (TFSB).

### 2.0 INTERFACES

There are two interfaces to the WD2840: the host computer side, and the network side. The network side is conventional from an electrical point of view, the WD2840 performs all logical functions required to ensure communications capability on broadcast media (such as coax or RF).

The host interface involves two separate functional interfaces: the status/control registers described in section one, and a DMA interface that is described in the following subsection.

#### 2.1 HOST

The WD2840 uses a complex memory buffer architecture allowing it to respond in real time to its network obligations (e.g., to meet network data rate and processing delay requirements). These memory structures are managed cooperatively by the host and the WD2840.

Memory management functions requiring real time response (e.g., traversing chains) are completely handled by the WD2840. Other important, but not time critical operations are the responsibility of the host software (such as removing used buffers from the transmit chain).

All memory references by the WD2840 are pointed to by memory locations (and internal registers) initially defined and set up by the host software. Initial values and memory based registers are grouped together and called the WD2840 Control Block.

The location of this control block is written into the registers CBPH and CBPL anytime the WD2840 is in Isolate State. This control block has the following structure:

CBP → +0	NXTR (H)	Receive Buffer Chain (MSByte)
+1	NXTR (L)	Receive Buffer Chain (LSByte)
+2	NXTT (H)	Transmit Buffer Chain (MSByte)
+3	NXTT (L)	Transmit Buffer Chain (LSByte)
+4	BSIZE	Buffer Size / 16 (0-F = 64-1024 bytes)
+5	EVT0	Eleven separate Event Counters, see section 2.1.1 for details
+6	EVT1	
.....		
+F	EVT10	

As the WD2840 transitions to Network State, it reads and uses the first five bytes of the control block. The remaining eleven bytes of event counters are accessed by the WD2840 only when each specific event condition occurs.

Either the Receive (NXTR) or Transmit (NXTT) chain entries in the control block may initially be zero; in such a case the WD2840 expects the chain to be extended by the host's changing the zero link field in the control block. Thereafter any such zero link would be in a buffer.

The WD2840 uses constant size buffers; their length is set by the value in location BSIZE. The buffer size is indicated by a 4-bit count in the least significant 4 bits of the BSIZE byte in the WD2840 control block. The buffer sizes available are multiples of 64; (BSIZE + 1) 64 is the buffer size used by the WD2840. Thus a BSIZE range of 0-15 corresponds to actual buffer sizes of 64 through 1024 bytes. This buffer length is inclusive of control bytes and buffer link pointers.

The WD2840 includes a chained-block feature which allows the user more efficient use of memory, particularly in situations where the maximum packet size is much larger than the average packet size. One or up to 16 buffers may make up a frame but a buffer may not contain more than one frame.

Byte counters are associated with each frame (at the memory interface, not actually transmitted within the frame) so that frames on the network need not be integer multiples of buffers. The byte counters include all buffer management overhead. Therefore, a frame consisting of 100 transmitted data bytes, occupying two 64-byte buffers, would have a byte count of 110 (six bytes per frame + 2 bytes per buffer).

Since the WD2840 receive and transmit buffer chains are linked lists (see section 2.1.2 and 2.1.3) and are "followed" by the WD2840 but managed by the host;

it is expected that the host will maintain both a FIRST and a LAST address for each chain. On transition into Network State, the chain origin information in the WD2840 control block is the same as FIRST. In fact, since the WD2840 does not change these control block entries, they can be maintained directly as FIRST by the host. An explicit LAST could be placed in an extended control block section.

The WD2840 "follows" the linked buffer chains by maintaining a NEXT address internally for each chain. This NEXT address can be in one of two states: 1) it can be the address of the next buffer in the chain, or 2) at the chain end (zero link), it can be the address of the buffer containing the zero link. The WD2840 uses a status bit for each chain, NXTR0 (receive) and NXTT0 (transmit), to differentiate the two states. When set they indicate the WD2840 chain NEXT address is in state 1 above; when clear they indicate state 2 above. This is an important distinction since it indicates whether the last buffer posted in a chain can be removed by the host (because the WD2840 has advanced to the buffer beyond) or must be left until the chain can be extended so the WD2840 can advance.

The host software monitors the progress of the NEXT pointer, and updates FIRST and LAST as it adds (and removes) buffers to (from) the chains as required. The WD2840 provides Interrupt Events (see IR0) and NXTR0, NXTT0 status bits to indicate when it advances along the two chains and exactly what state its NEXT address registers are in. The operation of these chains will be explained by example in later sections.

#### "Deadly Embrace" Prevention

A "Deadly Embrace" can occur when two processors reach a state where each is waiting for the other. In this case, the two processors are the user's CPU and the micro-controller inside the WD2840. Therefore, to prevent the "deadly embrace," the following rule is obeyed by the WD2840 and should also be obeyed by the user's CPU. This rule applies to the WD2840 memory registers and to the I/O registers. The Event Counters are an exception to this rule.

Rule:

If a bit is set by the CPU, it will not be set by the WD2840, and vice versa. If a bit is cleared by the WD2840, it will not be cleared by the CPU, and vice versa.

As an example, the NEWNA (CR10) control bit is only set by the host and is only cleared by the WD2840.

#### Dual DMA

The WD2840 may, for efficiency, interleave frame data fetch/store operations with fetches and stores of pointers and flags in memory. In all cases, operation sequencing is such as to prevent deadlocks and ambiguities between the WD2840 and software.

### 2.1.1 EVENT COUNTERS

Several non-fatal logical events are tabulated by the WD2840 and made visible to the host via memory based event counters (see WD2840 control block organization for specific locations). The WD2840 will

increment each counter at the occurrence of the specified event. Note that the WD2840 will not increment past 255. The host has the responsibility of initializing each counter.

COUNTER	DESCRIPTION
EVT0	"Set scan mode" frame received from the network. The NA register was redefined to MA + 1 at the time.
EVT1	Transmission error first attempt, second try successful. Can only occur for frames requiring an acknowledgement. It indicates no response was received for the first transmission; however, the second transmission was either ACK'ed or NAK'ed.
EVT2	Transmission error. Attempt aborted due to either transmitter underrun or frame length exceeding 16 buffers.
EVT3	Timer TD (network dead) expired.
EVT4	Access Control Frame Reception Error. A one or two byte supervisory frame (ACK/NAK, Token Pass, Scan Mode) has been received in error. This may be due to an FCS error, frame abort, or carrier loss detection.
EVT5	Data Frame Reception Error. An incoming data frame was incorrectly received due to an FCS error, frame abort, carrier loss detection, or receiving a data frame when expecting an ACK/NAK frame.
EVT6	NAK sent. Can occur for any of the following reasons: <ol style="list-style-type: none"> <li>1. Insufficient buffers in chain</li> <li>2. Receiver not enabled (RXEN clear)</li> <li>3. Receiver overrun</li> <li>4. Frame length exceeded 16 buffers</li> </ol>
EVT7	Invalid frame received. Caused by the detection of certain abnormal network conditions such as receiving an ACK/NAK frame when not expecting one, receiving a Scan mode frame when expecting an ACK/NAK frame, or receiving an invalid supervisory frame.
EVT8	Duplicate token detected. This counter will be incremented when the WD2840 determines that more than one token exists in the logical ring. This happens if a token pass is received when the WD2840 already has the token, or a data frame is received when the WD2840 is waiting for an acknowledgement frame.
EVT9	Not used.
EVT10	Duplicate node address. This counter will be incremented when a data frame being DMA'd into memory has a source address (SA) equal to the WD2840 node address (MA). This counter when used with COPY mode (CR02) is one way for detecting other nodes with the same node number (MA).

### 2.1.2 TRANSMIT MEMORY INTERFACE

When the token is received, data transmission is enabled (TXEN - CR06 and TXDEN - CR07 both set), and if the access hold-off counter has reached its limit, the WD2840 will determine whether any data frames are pending in the transmit chain. If so, it will transmit the first data frame in the chain. Otherwise the token will be passed. A given data frame will be the last frame transmitted for this token if any of several conditions occur:

1. ISOL (CR00) is set indicating the host has requested a transition to Isolate State.
2. TXDEN (CR07) is clear indicating the host has changed data frame transmission rights.
3. The frame FSB indicates this frame should be the last transmitted for this token.
4. The running frame counter has reached its limit (TXLT).
5. No further frames are pending in the transmit chain.

If any of the first four reasons above are true a token pass will occur. If the last frame does not require an acknowledgement, the WD2840 will piggyback the token pass if that is permitted (CR16). If the token cannot be piggybacked or if the last frame transmitted is the last frame pending (condition #5 above), an explicit token pass will occur. A piggyback token will not occur for the last pending frame because, for the general multiple buffer case, it is not known to be the last pending frame until after the transmission is complete.

The WD2840 will read and evaluate the address of the next frame at two specific points in time:

1. At the end of the prior frame, even if the prior frame is the last to be transmitted for this token.
2. When the token is received and data frame transmission is permitted.

If a non-zero frame address is found at time 1 above, it is kept and used without being re-read at time 2 above. However, if no pending frame is found at time 1, this is noted with the NXTT0 flag clear and the chain re-inspected on each occurrence of time 2 above.

As frame transmission commences, the WD2840 reads the address of the next buffer, the frame control byte, (FCB) and the frame length. It then starts reading bytes from the buffer and sending them until the frame length count or the end of the buffer is reached. The new buffer is read and data transmitted as before. (See Figure 2.1)

The frame length provided in the LENGTH field must be the sum of the overhead bytes and number of data bytes (see Fig. 2.1).

Simplified formula for LENGTH:

$$\text{LENGTH} = \# \text{ of data bytes} + 2 \text{ link bytes per buffer} + 6 \text{ overhead constants per frame.}$$

Example #1

$$\text{LENGTH} = 8 \text{ (0010H in LENGTH field)}$$

implies one buffer is used for this frame (64 bytes) two link + six overhead, no data.

Example #2

$$\text{LENGTH} = \# \text{ of data bytes} + 2 \text{ link bytes per buffer} + 6 \text{ overhead constants per frame.}$$

Programmed buffer size = 64 bytes per buffer. Two buffers are used in this frame for a total of four link bytes (2 per buffer), six overhead, and 57 data bytes.

The General Formula for LENGTH

WHERE

$N_D$  = # of data bytes (max 4095)

6 = Overhead Constant per frame (FSB, FCB, LENGTH (H), LENGTH (L), DA, SA)

$B_K$  = B SIZE in bytes (64, 128, etc.) a constant preprogrammed into the WD2840 on 64 byte boundaries to a max of 1024 bytes.

GIVEN  $N_D$

$$L = N_D + 8 + 2 * \text{TRUNC} \frac{N_D + 5}{B_K - 2}$$

GIVEN L

$$\#B's = 1 + \text{TRUNC} \frac{L - 1}{B_K}$$

$$N_D = L - 6 - 2(\#B's)$$

**NOTE:**

The expression for  $N_D$  fails for values of  $L = B_K + 1$ . This is okay since the 2840 doesn't generate such values.

Examples: Find L given  $N_D$

	$N_D$	L	
	0	8	
	1	9	1 bufr
	56	64	
$B_K = 64$	57	67	2 bufrs
	118	128	
	119	131	3 bufrs
	120	128	1 bufr
$B_K = 128$	121	131	2 bufrs
	246	256	
	247	259	3 bufrs

Find #B's,  $N_D$  given L

	L	#B's	$N_D$
	8	1	0
	9	1	1
$B_K = 64$	64	1	56
	67	2	57
	128	2	118
	131	3	119
	L	#B's	$N_D$
	128	1	120
	131	2	121
$B_K = 128$	256	2	246
	258*	3	246
	259	3	247

**\*NOTE:**

Case corresponds to buffer end and frame end on same byte . . . extra buffer consumed.

When the frame length is finally reached, the WD2840 pauses if an acknowledgement has been requested. The frame status byte (FSB) is updated when the frame is completed; its posting indicates frame completion and gives information about the success or failure of the frame transmission. At frame completion, the WD2840 attempts to advance along the transmission chain to identify the next frame regardless of whether it will be transmitted with this token or later.

The host may add frames to the end of the transmit chain at any time by changing the zero link in the last buffer. Also buffers of all posted frames up to but NOT including the last buffer of the most recently posted, may be arbitrarily removed from the chain. The last posted frame (more specifically, the last buffer of the last frame) may only be removed and reused if NXTT0 is set. This indicates that the WD2840 has advanced its NEXT address to the next frame but that its transmission has not been completed (in fact, perhaps not even started).

**NOTE:**

The WD2840 checks only the most significant byte of the link field for zero link detection. This has the following implications:

1. When writing into a zero link field, the host must write the LSB of the new link field first, followed by the corresponding MSB.

2. All buffers must have a starting address greater than or equal to Hex '0100'.

#### TRANSMIT FRAME STATUS BYTE (WRITTEN BY WD2840)

BIT #	7	6	5	4	3	2	1	0
Name	DONE	WIRING	X	X	SELF	VAL2 — —	VAL1 — —	VAL0

BIT	NAME	DESCRIPTION
7	DONE	Set to guarantee a non-zero value for the posted FSB.
6	WIRING	Value of the corresponding bit in received ACK frame.
5-4	—	Reserved.
3	SELF	When set, indicates the ACK/NAK code appears in the value field (bit 2-0) of this FSB is assigned by the WD2840 transmitter routine. When clear, indicates value resulted from ACK/NAK code from receiving station.
2-0	VAL	An encoded field whose interpretation depends upon the SELF flag (bit 3) in this FSB. <ul style="list-style-type: none"> <li>a. SELF clear               <ul style="list-style-type: none"> <li>0 0 0 — No receive error (= ACK when DONE is set).</li> <li>0 0 1 — Insufficient buffers for frame.</li> <li>0 1 0 — Receiver not enabled at frame start.</li> <li>0 1 1 — Receiver over-run.</li> <li>1 0 0 — Frame exceeded 16 receive buffers.</li> </ul> </li> <li>b. SELF set               <ul style="list-style-type: none"> <li>0 0 0 — No transmit error.</li> <li>0 0 1 — Transmission failed after retry.</li> <li>0 1 0 — Transmission under-run.</li> <li>0 1 1 — Premature end of chain.</li> <li>1 0 0 — Transmission frame exceeded 16 buffers.</li> </ul> </li> </ul>

#### Transmit Frame Status and Control Bytes

Each frame has two bytes reserved, one for host control information needed by the WD2840, the other for status information posted by the WD2840 at frame transmission completion. The frame control byte (FCB) is only read by the WD2840, never changed; the frame status byte (FSB), is written (posted) by the WD2840 with no regard for its prior contents. On completion, the FSB value will always be non-zero; it

is important that the host zero the FSB byte in order to be able to recognize a posted frame.

#### NOTE:

Specifically note in Figure 2.1 that the first buffer of each frame has a different structure than any overflow buffers for that frame. In particular, each frame has only one set of FSB, FCB, and LENGTH fields regardless of the number of buffers required by the frame.

#### TRANSMIT FRAME CONTROL BYTE (WRITTEN BY HOST)

BIT #	7	6	5	4	3	2	1	0
Name	WACK	FCBLF	TRANSP	X	X	X	X	X

BIT	NAME	DESCRIPTION
7	WACK	Wait for Acknowledgement. Instructs the WD2840 to wait for an ACK/NAK response from the receiver for this particular frame only. The token control (TC) byte in the frame is automatically set to cause the destination node to respond. This bit must NOT be set if the frame uses the broadcast destination address. Inadvertently doing so will cause the frame to be posted "Transmission failed, due to max retries."
6	FCBLF	Last Frame. This bit will cause the WD2840 to pass the token either piggybacked with this frame (if possible) or explicitly after the frame transmission completes.
5	TRANSP	Transparent Frame. This bit will cause the WD2840 to interpret the buffer contents to be the exact sequence of bytes to be transmitted. The normal token control (TC) byte and source address (SA) byte generation is suppressed. Note that for a non-transparent data frame the TC byte must NOT appear in the buffer.
4-0	—	Reserved.

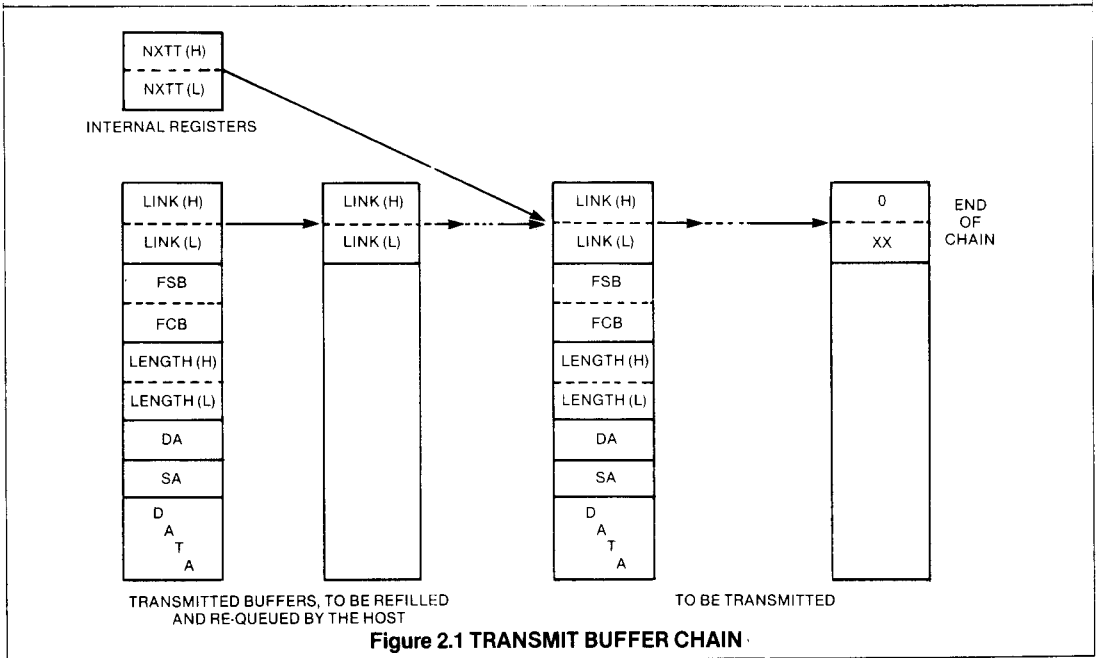


Figure 2.1 TRANSMIT BUFFER CHAIN

### 2.1.3 Receive Memory Interface

After the third byte of an incoming data frame is detected, the WD2840 will begin to place frame data into memory if several conditions are satisfied:

1. Receiver Enabled (RXEN-CR05 set).
2. There is an available buffer in the receive buffer chain.
3. The frame is addressed to this node specifically, it is a broadcast frame, or COPY mode has been selected by the host.

As the frame continues, it may completely fill its buffer. If this happens the WD2840 reads and inspects the link field of the current buffer. If this link is zero, an error occurs and the receive chain is reset to reuse from the first buffer used by the dropped frame. However, if another buffer is available, the incoming frame is continued beginning in the third byte of that buffer. This continues until one of several things happen:

1. Receiver overrun. The WD2840 has a four byte FIFO to buffer incoming frame data; however, if the host DMA responds too slowly a receiver overrun will occur. If this happens an event counter is incremented, the frame is dropped, and the receiver buffer chain is reset to reuse buffers of the dropped frame.

2. Current buffer capacity exhausted. If 16 buffers have been used for the current frame, an event occurs with the frame being dropped and the chain reset. Otherwise the WD2840 attempts to advance to the next buffer in the receiver buffer chain. The frame data will be continued in this subsequent buffer. If the end of the receiver buffer chain is reached an event counter is incremented, the frame is dropped, and the chain reset.
3. Frame ends. If the FCS is not correct an event counter is incremented, the frame is dropped, and the chain is reset. If correct however, the frame length is placed in the LENGTH field and the Frame Status Byte (FSB) is posted "done, no error."

If the frame is addressed to this node and indicates an acknowledgement is required (TC = 255), whether or not an error occurs, the WD2840 responds with an ACK/NAK supervisory frame indicating either success or failure. In case of receiver over-run, bad FCS, and SA = MA acknowledgement request will be ignored. (See section 1.4.3 for details)

It is the host's responsibility to ensure that buffers are available, initialized (FSB zero'ed), and attached to the end of the receive buffer chain.

**RECEIVE FRAME STATUS BYTE (WRITTEN BY WD2840)**

BIT #	7	6	5	4	3	2	1	0
Name	DONE	X	X	X	X	X	X	X

BIT	NAME	DESCRIPTION
7	DONE	Set to indicate the frame reception is complete.
6-0	—	Reserved.

**RECEIVE FRAME CONTROL BYTE (WRITTEN BY HOST)**

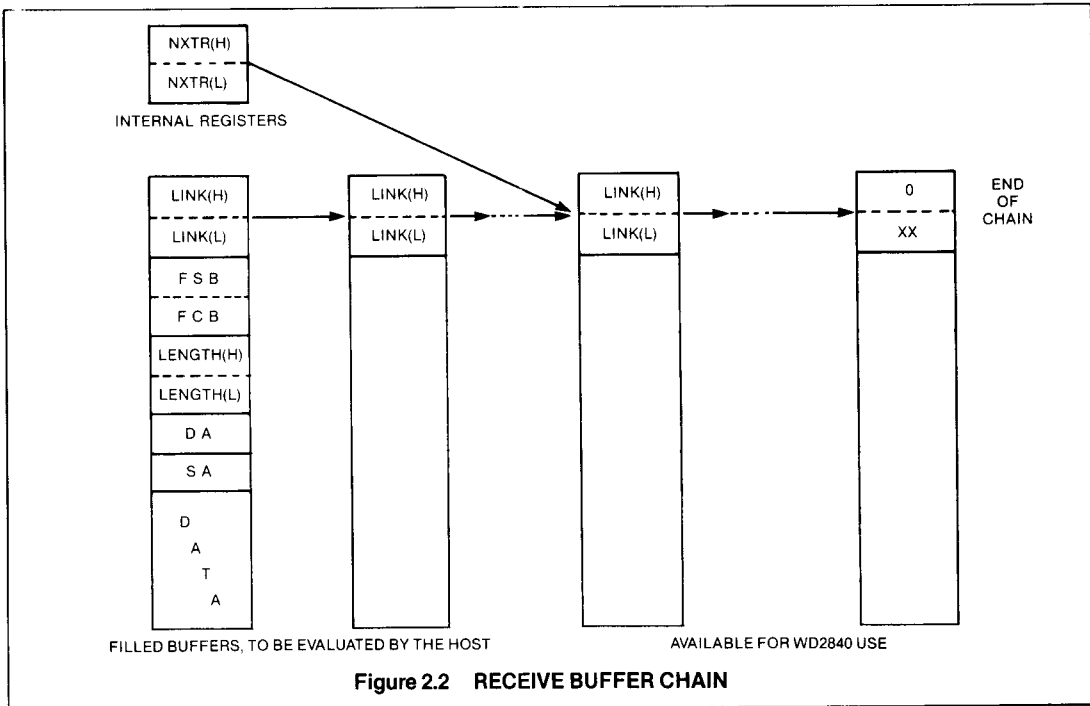
BIT #	7	6	5	4	3	2	1	0
Name	X	X	X	X	X	X	X	X

BIT	NAME	DESCRIPTION
7-0	—	Reserved.

**2.2 MODEM INTERFACE**

The modem interface is the conventional half duplex NRZ type with separate data and clock (Figure 2.3). When the WD2840 desires to transmit, it asserts RTS and awaits CTS. RTS is generally used to enable the modem transmitter. After a system dependent preamble is generated, the modem asserts CTS which allows the WD2840 to begin the actual transmission of the frame. (Note: CTS may be asserted permanently if the transmission system does not need to generate a preamble).

The SQ input is used on receive to indicate a valid carrier. If this term is negated anytime during a receive message, the WD2840 will presume the message is in error and treat it as an abort. This signal is used to augment message integrity beyond that of the CRC by allowing a modem to detect and report low level faults (such as out-of-frequency carrier or missing clock).



**Figure 2.2 RECEIVE BUFFER CHAIN**

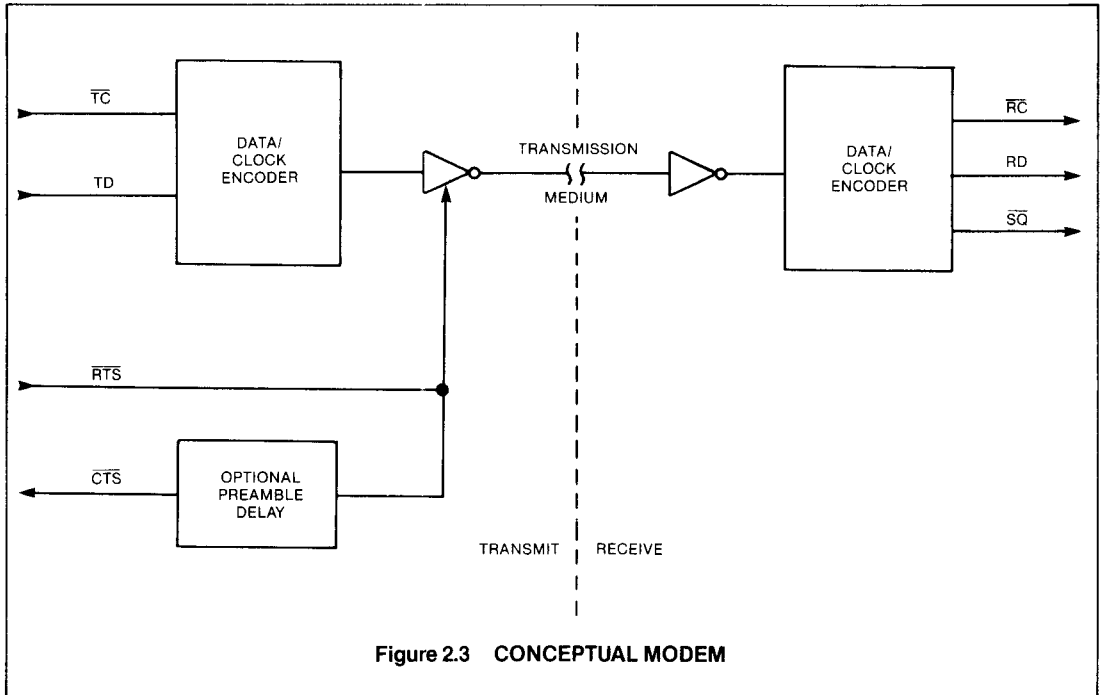


Figure 2.3 CONCEPTUAL MODEM

### 3.0 NETWORK PROTOCOL

To enable operation on a broadcast medium without the need for a central controller performing device polling, the WD2840 implements a media access protocol. The particular access protocol designed into the WD2840 prevents self-induced transmission collisions and ensures a fair and guaranteed distribution of transmission time among attached controllers. (See Appendix A for Protocol flowcharts.)

This design-out of collisions allows the WD2840 a greatly expanded selection of transmission media, since no physical characteristics of a particular medium are relied upon for proper network operation. Another benefit of this lack of collisions is the visibility of network faults. If a collision is detected, it is treated consistently in a error recovery mode by the WD2840 and is also unambiguously visible to service personnel as a fault.

Secondly, the WD2840 can ensure that a transmitted message was correctly received and buffered by requiring acknowledgement of its receipt. This is sometimes called "acknowledging datagrams" where the sender awaits a predefined period after a frame is sent for a reply from its destination. With this method, no sequence counters nor multi-frame retransmission buffering is required. The scheme is efficient since local network applications such as the WD2840 address do not encounter extremely long transmission delays (such as satellite links) as in conventional data networks (such as X.25).

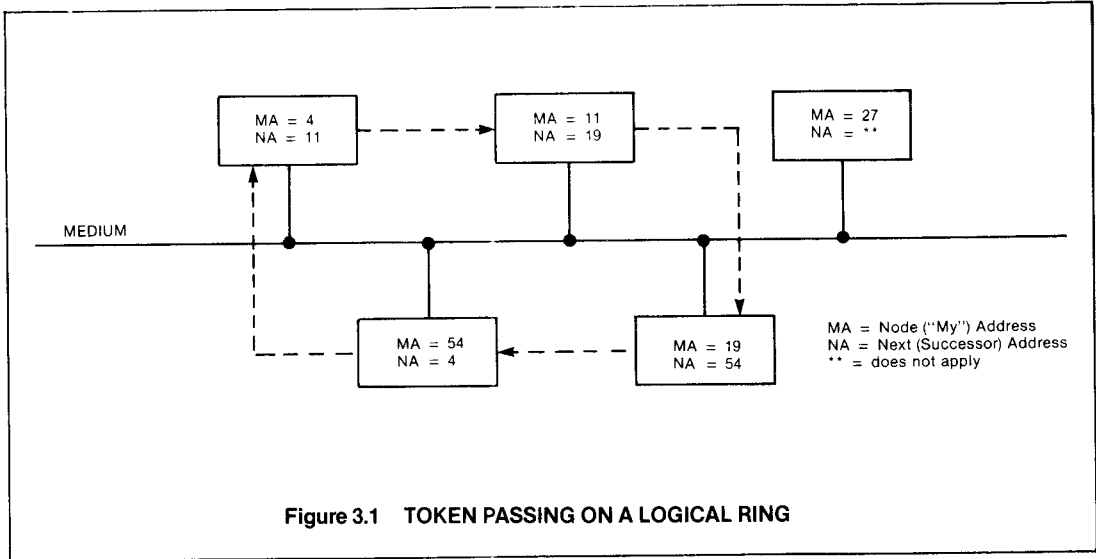
Both functions are parameterized, allowing tuning and optimization by the user to his unique application. These parameters may be adjusted in real time by the user's software, allowing a dynamic network, responsive to constantly changing requirements.

The two functions, access control and data transmission, function simultaneously though independently. Thus they are described separately as subprotocols for clarity.

#### 3.1 Data Transmission

The data transmission cycle is entered after the token has been received and data transmission rights validated (see section 3.2 "access method"). The WD2840 determines if there is a frame to be sent and, if not, simply sends the token to the next station.

If something is queued for transmit, the WD2840 DMA's it from memory and sends it. After the complete frame has been sent, the WACK (Wait for ACK) bit is tested in the TFSB (Transmit Frame Status Byte). If set, the WD2840 waits for, and expects, an acknowledgement from the frames recipient. A timer (TA) is started. In the normal case, the ACK is received before TA expires which causes the WD2840 to send the next frame queued, repeating this procedure. Thus, the WD2840 sends multiple frames to various destinations until the transmit queue is emptied or a programmed limit (register TXLT) is exceeded.



In the event TA expires, the frame is re-transmitted once. (Note: it is the responsibility of higher level protocol operating in the host to protect against the possibility of duplicate frame reception.) If TA expires again, usually indicating the destination node is off-line, the FSB is updated to reflect the unsuccessful transmission, interrupt bit ITA is set, and the frame is skipped.

A frame is also skipped and tagged if the destination station sends a NAK, indicating it cannot presently process the frame.

### TRANSMISSION OF ABORT

An ABORT is transmitted by the WD2840 to terminate a frame in such a manner that the receiving station will ignore the frame. An ABORT is sent when there is a Transmitter Under-Run. The abort sequence is a zero, followed by seven ones, after which RTS is set false.

### 3.2 ACCESS METHOD

The WD2840 network access method is based on the use of tokens, the specific granting of transmission rights passed from station to station. At any given time, exactly one station has the right to transmit (this right is called the token) and is obligated to pass it on when finished with it.

This can be clarified by referring to Figure 3.1. We assume in this figure that the network has already been initialized (meaning that the linkages in the access ring have already been established) and the

token is held at this instant by station 4 (the station whose MA register = 4).

When station 4 is ready to pass his access right on, he sends a message to the station number called out in his internal register NA, in this case 11. The message, and thus the token, are received by station 11 who can now transmit its message(s). When station 11 is ready to pass the token, it sends a message to station 19, as directed by its internal register NA and the cycle continues, in a circular fashion, from station 4 to 11 to 19 to 54 to 4 . . .

Notice that the station numbers need not be contiguous. This relatively arbitrary station numbering (in the example) poses no inefficiency to the access method. The value of this is the ability to add and remove stations (re-configure) on the network without re-arranging everyone else's addresses. (See section 3.2.2 for an example).

In this way, the token is passed from one station to the next in a logical ring.

#### 3.2.1 ACCESS INITIALIZATION/ERROR RECOVERY

When the WD2840 is commanded into Network State, the Next Address Request (NAR) register and the NEWNA (CR10) flag must be used to define the Next Address (NA) register. When it is necessary to pass the token, it is passed to the current node number in register NA. If station NA is not on-line, determined by its lack of response, station NA + 1 is tried. This process continues until a station is found

which does respond. The responding station number is written into register NA so that this scanning procedure need not be repeated on subsequent access cycles.

NOTE: 1. Node numbers 0 and 255 are reserved and cannot be used. Consequently scanning occurs circularly in the range 1-254.

2. During Scan mode token passing to each node is only tried once.

Anytime a station cannot successfully pass a token within two attempts, register NA is updated to NA + 1, and a new "next" station is searched for. The result is the removal of non-responding station(s) from the access ring. An interrupt (INS) is generated indicating a network exception caused a change to NA.

The above description covers network recovery from station failure and purposeful removal of stations during on-line network operation. Setting stations in the scan mode can also be accomplished by sending control frames (a Scan frame redefines NA = MA + 1) over the network. The control frame may be directed to a single station, or all stations simultaneously (using the broadcast address). It is this scanning for new stations that permits on line addition to the access ring.

NOTE:

The policy of the SCAN frame is redefined by the user software as required by the application. For example: in a process control environment where stations are not often added while the network is in use, this procedure would be initiated rarely if at all.

### 3.2.2 REMOVING A STATION

There are two ways a station can be removed from the access ring: non-response due to station failure and non-response due to host commanded transition to the Isolate State. Both are treated identically from a network point of view.

Referring to Figure 3.1, assume that station 19 is removed from the network (either physically or logically). In this example, station 11 would detect a network fault when trying to pass the token to 19 (time TA would expire since station 19 will not respond). Station 11 detects this and finds the next station in the access ring by using the "scan" function (similar to initialization). The next attempt at passing the token would be to station 20, register NA + 1.

By starting the token ring recovery procedure at the intended station plus one (station 20) rather than MA + 1 (station 12) as is done in initialization, recovery delays are minimized (since fewer stations are tested for presence, 8 less in this example).

The next station found would be number 54 in the example which station 11 writes into his register NA (now "patching out" dead station 19). The next time station 11 is finished with the token, it directly sends it to 54, making the sequence now 11 to 54 to 4 to 11 to 54...

### 3.2.3 ADDING STATIONS

There are three primary methods by which a station can be added to a network. The first is a distributed method, in which each station in the network can poll for new stations in the gap between its address and the next address (between MA and NA). Second is a centralized method, in which an individual station designated by the network architect can interrogate the entire address space seeking a new station desiring INRING. The third — central scan — is a simpler (from the host point of view), centralized method in which a station can send a global frame causing all the on-line TACs to reset their next address register. This causes each TAC to poll its address space at its next token-pass attempt. Each method has advantages and disadvantages.

#### Distributed Method

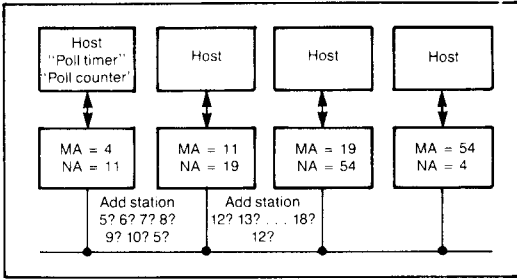
The distributed method does not rely on a specific station. Thus, there are no problems or efforts spent selecting the administrator, nor is there any concern about backup administrators. In the distributive method, each station has the same responsibility to allow new access members as other stations. This method is the most host intensive and requires each station to maintain a timer (that can be configuration set as to its value) as to how often it should poll its gap for new stations.

For example, assume the timer in each station is 5 sec. and that station 4's timer has expired (Fig. 3.2.3.1). The host attached to station 4 notes that the next address register (NA in the TAC) is set to 11, which indicates that a new station might be added to the network as station number 5, 6, 7, 8, 9, or 10.

The host queues a frame into the TAC transmit chain, polling station 5. This frame will be sent by 4 with an acknowledgement requested from 5. If 5 is present it responds; otherwise, the TAC aborts its attempt after time TA. The TAC marks the result on the frame in the host memory space and proceeds with other tasks.

After this exchange, the host, at its leisure, checks the transmit status of the frame. The host sees that the frame acknowledgement timed out, meaning that station 5 has not been added to the network, or that station 5 is on the network and whether the request INRING is set in the network code field. In either case, the host takes appropriate action. If the desired INRING bit is set, station 4 changes its NA register to 5, allowing its next token to be passed to 5. This action puts station 5 in the ring.

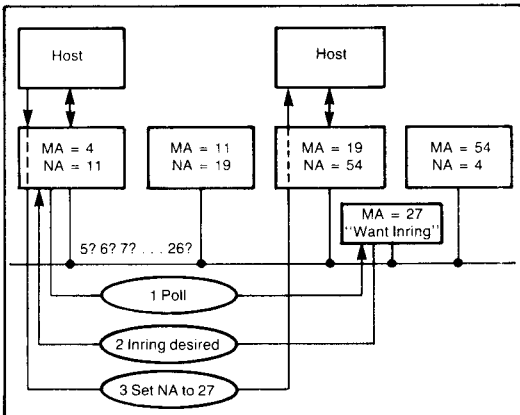
Depending on an application's sophistication, a control message can be sent to station 5. That message says, "Your successor is X." In this case, X = 11, so that 5 is not forced to poll for its successor. In any case, 4 updates its next address register to 5 and does not need to go through this distributive polling cycle again because there is no gap between 5's address and the next address; there is no possibility that a new station can be inserted between addresses 4 and 5. If 5 didn't respond to 4's poll, station 4 updates its poll counter so that the next time that the poll timer times out, station 6 will be tried.



**Fig. 3.2.3.1 Distributed polling.** Each host polls the gap in its address space for the possible addition of new stations. The host internal poll timer and poll counter set the polling rate and range as desired.

If node 6 responds, its desired INRING bit is tested as above. If 6 does not respond, the host will queue a poll to station 7 the next time its poll timer expires. This continues until the host completes 10, when the cycle goes back to 5 and repeats. In this example, with a gap of 6 stations (between 4 and 11), and with a 5-sec. clock, a new node can be added within 30 sec.

In the centralized station-addition method, a single station can poll the entire address space, seeking a new station that desires INRING. One reason for centralizing this function might be the more careful control that can be placed in a network. There can also be optimizations. For example, the central polling station can keep track of the stations that already exist and, therefore, bypass some address ranges. A polling station may know the network will never have more than, say, 75 stations. In the example of Fig. 3.2.3.2, when station 4 starts polling, it polls only to address 75 before resetting to zero. This works like the distributed method except that a single station does all the work.



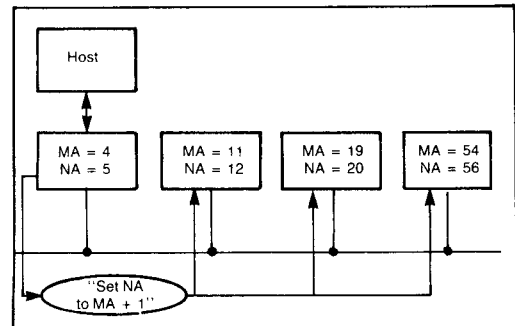
**Fig. 3.2.3.2 Central polling.** A single station — in this case, station 4 — dubbed “the administrator,” can be charged with all polling tasks. This simplifies the software in the other stations and centralizes network control.

When the polling station determines that a station has been added, it must place the new station in the access ring. For example, station 4 is the centralized station doing all the polling (Fig. 3.2.3.2), and it discovers that station 27 has recently been added. Station 4 knows this because station 27 now responds to a first-time poll, and because its status bit is set, indicating that it wants to be added to the ring. (Some stations may be receive only, never desiring the right to initiate transmissions.) Station 4 sends a high-level message to the software in station 19, telling it to change its next address register to 27. This message can also prompt station 19 to tell 27 its next address register should be 54. This gets confusing, but it is all done with high-level software. These tasks are not real time and are quite efficient from the network point of view.

Station 4, the administrator, need not create and maintain a table of active stations on the network because the poll response returns three pieces of information. As node 4 polls the stations on the network, it finds out (a) that the polled station does not respond at all, as it would if it polled station 12 in Fig. 3.2.3.2 (b) that the station is already part of the network and is already in the ring or is receive only, as it would if station 4 happened to poll station 11 or 19; and (c) whether the station is attached to the network, is alive and wants to be in the ring, as is the case with a poll to 27. These indications are conveyed by a combination of status bits sent back by the acknowledge frame. This acknowledge frame and status information are transferred at a TAC device level, so a host is not concerned with whether its station wants to be in the ring. The host simply sets up the proper bits in the control registers; the bits are relayed automatically by the TAC. Thus, with a simple algorithm, an administrative station can poll the entire network address range and know the network’s exact membership and status.

**Central Scan**

Central scan is the simplest method of adding stations to a network. It involves sending a global frame to all



**FIG. 3.2.3.3 Central scan request.** A special command can be sent by any station causing all attached TACs to set their NA register to the address of the next possible node. This causes each TAC to poll without the help of the host.

stations on the network, which forces each to update its own next address register to its station address plus one ( $NA = MA + 1$ ). Assume station 4 is the centralized station and sends the scan command frame (Fig. 3.2.3.3). Station 11, upon receiving it, automatically sets its next address register to 12 (the TAC does this; the host is not involved but is notified of the situation). Also, station 19 sets its next address register to 20, and station 54 sets its NA register to 55.

The result of this is a round of polling at the TAC level. Station 11, on completing its use of the token, tries to send it to 12. The token to station 12 times out because 12 is not present. Station 11 reclaims the token trying to send it to 13 and so on, causing 11 to poll for station addition. The drawback of this is the huge time disruption incurred by the simultaneous polling.

It is not required that station 4 send this scan control frame to all stations at the same time. If it is known that station 11 exists in the network and that a station may be trying to add into the network after station 11 in the address space, a command can be sent to 11 telling it to set its next address register to  $11 + 1$ . Now 11 will go through scanning station 12, 13, 14 . . . again without intervention from station 11's host software. This directed scanning has the effect of smoothing the polling disturbance over a greater time.

The trade-off of all these methods is the software complexity distribution. If a TAC user assumes more responsibility, providing more intelligence distributed in the software, the system can be more sophisticated in handling new stations. If a user wants the TAC to handle this task itself, saving host software development, he pays only slightly in inefficiency. TAC gives the user an option.

**3.2.4 INTERACTION OF THE SUB-PROTOCOLS**

After a station is given the token, it will send an information frame, a token frame, or a combination of both. It is this combination frame, referred to as a "piggy back" token, that causes the sub-protocols to interact slightly.

In the normal case (no time-out), the SOURCE may transmit a combination frame to the DATASINK when his access period is over. All stations on the network observe this; after the reception of the current frame is complete, the one whose MA register matches the token address in the frame (TC) knows it has the token.

In the case of a combination frame, the SENDER resets his timer TA on transmission complete and waits for the NA station to transmit something valid,

**FIELD DESCRIPTIONS AND ENCODING**

TC	<p>The token control byte has the dual purpose of transferring access control between stations and conveying a request for immediate acknowledgement of the frame by its intended receiver.</p> <p>There is no interaction between the TC field and the DA or SA fields. Thus the token may be transferred to one station and data sent to the same or a different station, with one single frame. The value entered into the TC field is determined by the WD2840 and does not appear in the buffer (except for transparent frames).</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th style="text-align: center;">TC Value</th> <th style="text-align: center;">Meaning</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td>Token not affected at this time.</td> </tr> <tr> <td style="text-align: center;">1-254</td> <td>After current frame, the token belongs to station TC. (The sending station has recovery responsibility).</td> </tr> <tr> <td style="text-align: center;">255</td> <td>Immediate ACK requested. Token not affected.</td> </tr> </tbody> </table> <p>NOTE: The sharing of this field prevents the passing of the token with data (piggy-back) and acknowledgement requests on the same frame. This combination is specifically disallowed because of its undesirable characteristics in network error situations.</p>	TC Value	Meaning	0	Token not affected at this time.	1-254	After current frame, the token belongs to station TC. (The sending station has recovery responsibility).	255	Immediate ACK requested. Token not affected.
TC Value	Meaning								
0	Token not affected at this time.								
1-254	After current frame, the token belongs to station TC. (The sending station has recovery responsibility).								
255	Immediate ACK requested. Token not affected.								
DA	Destination address. Value of zero is reserved, 1 to 254 indicates the destination address of the frame. The value 255 is the global (or broadcast) address.								
SA	Source address. The values of 0 and 255 are reserved. A value of 1 thru 254 is the address of the sender of the frame.								
I	Information Field. User defines format and content.								
FCS	Frame Check Sequence. The FCS calculation includes all data between the opening flag and the first bit of the FCS, except for 0's inserted for transparency. The sixteen bit FCS is compatible with the standard HDLC FCS.								
AC	Access Control. Conveys supervisory information. May be sent as a command using transparent mode or received in response to an ACK/NAK request. Its format is shown below:								

## ACCESS CONTROL FIELD

BIT #	7	6	5	4	3	2	1	0
Name	SCANF	WIRING	0	0	0	NVAL2 — NVAL1 — NVAL0		

BIT	NAME	DESCRIPTION
7	SCANF	Scan Mode (Command). Indicates that the addressed node(s) must redefine NA = MA + 1 for use on its next token pass.
6	WIRING	Wants in ring (Response). This bit when set informs the node requesting the ACK frame that this node is not in the logical ring, but is requesting entry. It is the logical function of the transmitting node's GIRING .AND. INRING. (see CR13 and SR20) The WD2840 does not act on this information but merely passes it to the host via the ACK'ed frame's FSB.
5-3	—	Reserved.
2-0	NVAL	An encoded NAK/ACK value (Response). The receiving node will set one of the following codes depending upon the state of the last received frame: 0 0 0 — No error 0 0 1 — Insufficient buffers for frame 0 1 0 — Receiver not enabled at frame start 0 1 1 — Receiver overrun 1 0 0 — Frame exceeded 16 receive buffers

to verify his reception of the piggy back token. If the timer expires, the sender sends an explicit token (the data from the combination frame is assumed to have been accepted) and enters the normal token sub-protocol.

The user is prevented from sending a combination frame and requesting an acknowledgement at the same time to prevent possible network state conflict under time-out conditions.

### 3.3 FRAME FORMAT

The frame format the WD2840 uses to transmit all data and control frames is similar to the industry standard HDLC. A 16 bit CRC is implemented and standard zero insertion (CRC16-CCITT) is used for framing. This framing method allows the use of standard network monitoring and diagnostic equipment such as data scopes and logic analyzers.

Additional address fields and control points are defined as required to support the protocol.

#### Normal Frame Format:

F - TC - DA - SA - I - FCS - F

F = Flag, binary pattern 01111110  
TC = Token Control (8 bit)  
DA = Destination Address (8 bit)  
SA = Source Address (8 bit)  
I = Information Field (0 to 4095 bytes or 16 buffers, whichever is less).  
FCS = Frame Check Sequence (16 bit)

#### Access Control Format:

F - DA - AC - FCS - F

F = Flag, binary pattern 01111110  
DA = Destination Address (8 bit)  
AC = Access Control Field (8 bit)  
FCS = Frame Check Sequence (16 bit)

#### Token Pass Format:

F - TC - FCS - F

F = Flag, binary pattern 01111110  
TC = Token Control (8 bit)  
FCS = Frame Check Sequence (16 bit)

### 3.4 SENDING A TRANSPARENT OR ACCESS FRAME

Two types of frames are transmitted under the transparent mode under user control. A scan access frame or a transparent frame. The format of the frames are described under 3.3 Frame Format with the transparent format the same as Normal Frame Format.

#### ACCESS FRAME COMMANDS

There is only one Access Frame type permitted under user control — Scan Frame. The node that is addressed must redefine NA = MA + 1 for use on its next token pass. The format for sending this frame is:

LINK (H)	Pointer to next frame.
LINK (L)	
00 (FSB)	
20h (FCB)	Transparent frame, no acknowledge allowed.
00 LENGTH (H)	Access Control frame size (H, L).
08 LENGTH (L)	
DA	Destination Address or 255 broadcast.
80h	Set Scan Mode.

The FCB can be set for last frame, the acknowledge bit has no effect and no acknowledgements will be given to access frames nor will they be expected by the transmitting WD2840.

The node receiving the access frame will only recognize a scan access frame as a command. Event count #0 will increment and the receiving node will set its NA to MA + 1. Any other access code will increment Event Counter #7.

#### OPERATING CHARACTERISTICS (DC):

Operating Temperature Range . . . . . 0°C to +70°C

SYMBOL	PARAMETER	MIN.	TYP.	MAX.	UNIT	CONDITIONS
I <sub>DD</sub>	V <sub>DD</sub> Supply Current		18	30	mA	
I <sub>CC</sub>	V <sub>CC</sub> Supply Current		160	220	mA	
V <sub>DD</sub>	High Voltage Supply	11.4	12	12.6	V	
V <sub>CC</sub>	Low Voltage Supply	4.75	5	5.25	V	
V <sub>IH</sub>	Input High Voltage	2.4			V	
V <sub>IL</sub>	Input Low Voltage			0.8	V	
V <sub>OH</sub>	Output High Voltage	2.8			V	I <sub>O</sub> = -0.1mA
V <sub>OL</sub>	Output Low Voltage			0.4	V	I <sub>O</sub> = 1.6mA
I <sub>OZH</sub>	Three-State Leakage			50	μA	V <sub>IN</sub> = V <sub>CC</sub>
I <sub>OZL</sub>	Three-State Leakage			50	μA	V <sub>IN</sub> = 0.4V
I <sub>IH</sub>	Input Current			10	μA	V <sub>IN</sub> = V <sub>CC</sub>
I <sub>IL</sub>	Input Current			10	μA	V <sub>IN</sub> = 0.4V

### TRANSPARENT FRAME

Link (H)	
Link (L)	
FSB	
20H FCB	
XX Length (H)	
>08H Length (L)	
TC	pass token, broadcast, ackreq.
DA	
SA	
DATA	

The Transparent Data Frame allows a user to control the token pass, or TC field of a frame by using the first byte after length rather than the FCB. The frame transmitted will look like the User Info part of the buffer without the WD2840 firmware generating anything else but the flags and FCS.

### 4.0 ELECTRICAL SPECIFICATIONS

#### ABSOLUTE MAXIMUM RATINGS:

Voltages referenced to V <sub>SS</sub>	
High Supply Voltage (V <sub>DD</sub> ) . . . . .	- 0.3 to 15V
Voltage at any Pin . . . . .	- .03 to 15V
Storage Temperature Range . . . . .	- 55°C to + 125°C
Electro-static voltage at any pin . . . . .	400V (Note 6)

#### NOTE:

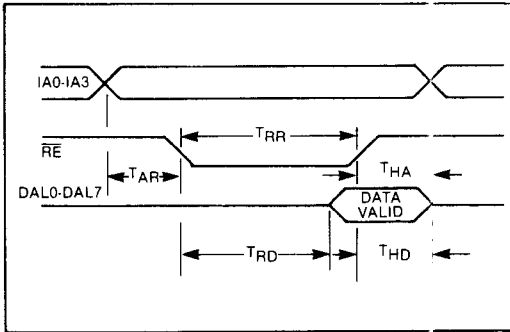
Absolute maximum ratings indicate limits beyond which permanent damage may occur. Continuous operation at these limits is not intended and should be limited to those conditions specified under DC Electrical Characteristics.

## 5.0 TIMING CHARACTERISTICS (AC):

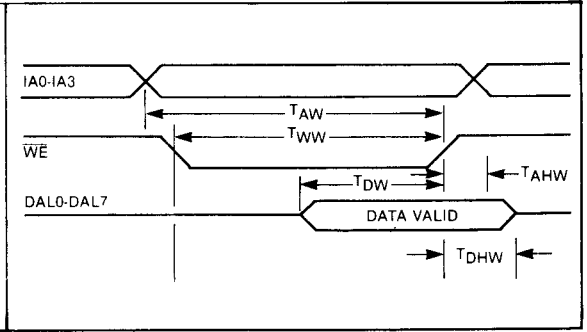
SYMBOL	PARAMETER	MIN.	TYP.	MAX.	UNIT	CONDITIONS
CLK	Clock Frequency	0.5		2.05	MHz	Note 1, 7
RC	Receive Clock Range	0			MHz	Note 4, 7
TC	Transmit Clock Range	0			MHz	Note 4, 7
MR	Master Reset Pulse Width	10			mS	
TAR	Input Address Valid to $\overline{RE}$	0			nS	
TRD	Read Strobe (or $\overline{DACK}$ Read) to Data Valid	2		375	nS	Note 5, 2
THD	Data Hold Time From Read to Strobe	20		100	nS	
THA	Address Hold Time From Read Strobe	0			nS	
TAW	Input Address Valid to Trailing Edge of $\overline{WE}$	100			nS	
TWW	Minimum $\overline{WE}$ Pulse Width	200			nS	
TDW	Data Valid to Trailing Edge of $\overline{WE}$ or Trailing Edge of $\overline{DACK}$ for DMA Write	100			nS	Note 2, 3
TWRR	CS High Between Writes	300			nS	
TRDR	CS High Between RE	300			nS	
TRR	RE Pulse Width	375			nS	
TDAK	$\overline{DACK}$ Pulse Width	375			nS	
TAHW	Address Hold Time After $\overline{WE}$	80			nS	
TDHW	Data Hold Time After $\overline{WE}$ or After $\overline{DACK}$ for DMA Write	100			nS	
TDA1	Time From $\overline{DRQ0}$ (or $\overline{DRQ1}$ ) to Output Address Valid if $ADRV = 1$			80	nS	
TDA0	Time From $\overline{DACK}$ to Output Address Valid if $ADRV = 0$			375	nS	Note 5
TDD	Time From Leading Edge of $\overline{DACK}$ to Trailing Edge of $\overline{DRQ0}$ (or $\overline{DRQ1}$ )			375	nS	Note 5
TDAH	Output Address Hold Time From $\overline{DACK}$	20		100	nS	
TDMW	Data Hold Time From $\overline{DACK}$ For DMA Read	20		100	nS	Note 2
TTDV	TD Valid	100			nS	
TSRD	RD Setup	0			nS	
THRD	RD Hold	320			nS	

## NOTES:

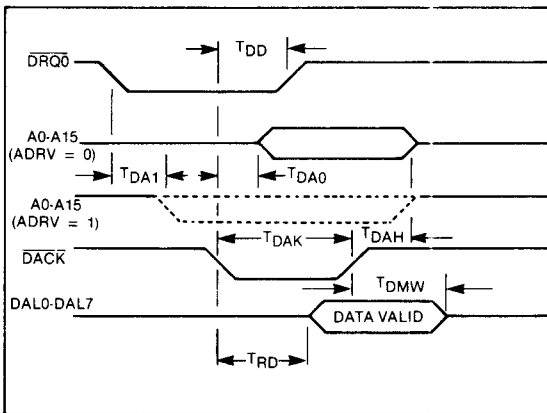
1. Clock must have 50% duty cycle.
2. There must not be a CPU read or write ( $\overline{CS-RE}$  or  $\overline{CS-WE}$ ) within 500 nanoseconds after the trailing (rising) edge of  $\overline{DACK}$ .
3. There must not be the leading (falling) edge of  $\overline{DACK}$  allowed within 500 nanoseconds after the completion of a CPU write ( $\overline{CS-WE}$ ).
4. See "Ordering Information" for maximum serial rates.
5. C(load) = 100pf
6. Measured by discharging a 100pf capacitor to each pin through a 1K ohm resistor.
7. TC/RC must be <43% of CLK when transmitting multiple buffers.



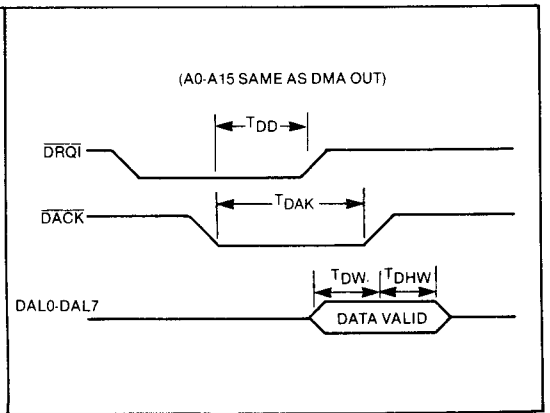
CPU READ ( $\overline{CS}$  IS LOW)



CPU WRITE ( $\overline{CS}$  IS LOW)



DMA OUT

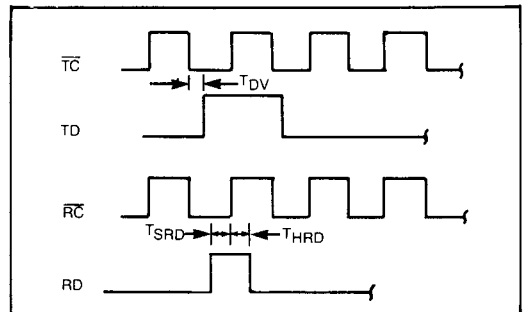
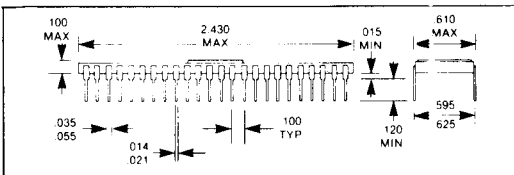


DMA IN

6.0 ORDERING INFORMATION

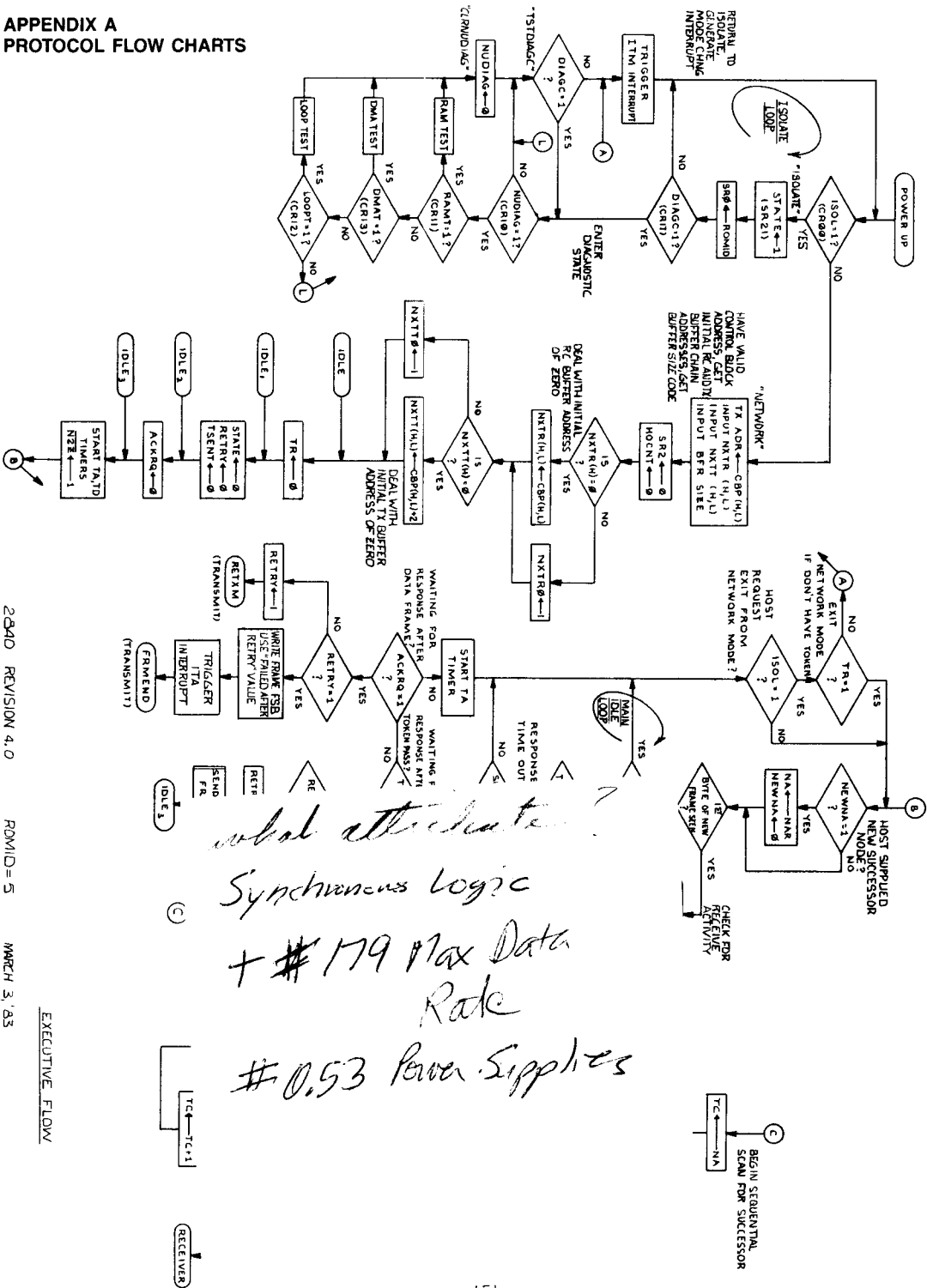
DEVICE NUMBER	MAXIMUM RATE
WD2840-01	100 Kbps
WD2840-05	500 Kbps
WD2840-11	1.0 Mbps

Package Diagram



TD-RD TIMING

APPENDIX A  
PROTOCOL FLOW CHARTS



2840 REVISION 4.0

ROMID=5

MARCH 3, '83

EXECUTIVE FLOW











## WD2840 Application Note

### INTERFACING THE WD2840 TO A VERSABUS SYSTEM USING A DUAL PORT MEMORY

This application note describes one possible method of interfacing the WD2840 Token Access Controller with a general purpose microprocessor based system. This is intended to be an example design only, no effort has been made to minimize the logic required to perform these functions as would be done in a production design. Rather, the design has been kept "clean" to promote readability.

This implementation is designed with a dual-port memory concept allowing its use in systems that either do not support DMA at all, as well as systems that are unable to guarantee reasonable DMA response to a request (Figure 1). Examples of these systems are low end personal computers that allow their disk controllers to "hog" the DMA channel for an entire sector transfer. Very high end systems are also candidates for the dual-port memory technique. Here, the system bus may be shared by multiple hosts and be of such extreme bandwidth that the internal WD2840 DMA controller may be inefficient.

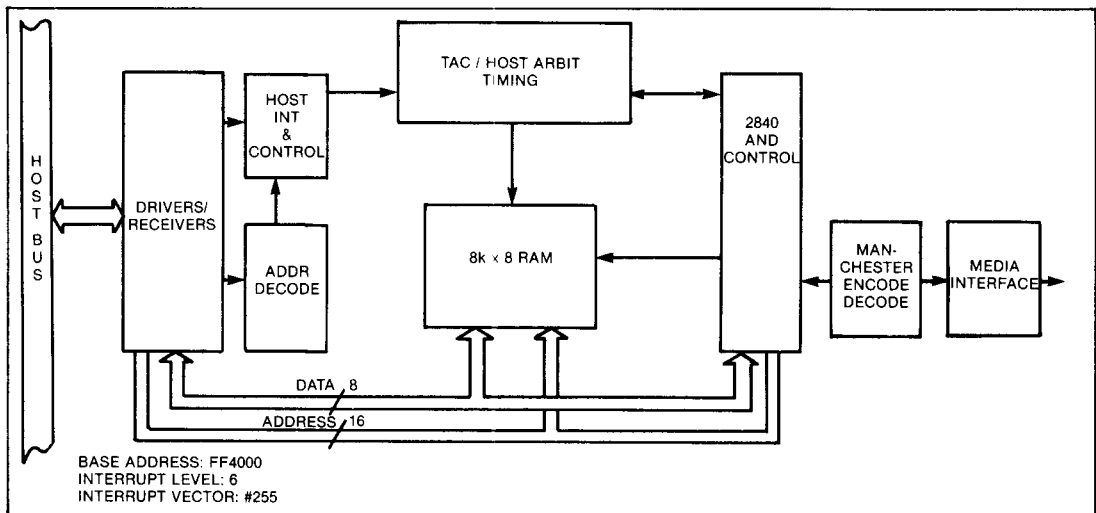
In most applications, the WD2840, can simply self-DMA its data directly to/from the host system's working memory. In the applications described above, the WD2840 must DMA its messages to/from the network into a local RAM allowing the host to access it at its leisure.

### REFERENCES

- WD2840 Token Access Controller Specification, Western Digital Corporation, 1983.
- "Token Passing Cashes in with Controller Chip" Electronic Design, October 14, 1982.
- Versabus Technical Reference Manual Motorola,
- RS-422 Technical Specification

This design is described in six sections:

- Host dependent logic, here designed for the Motorola Versabus system, including all required bus interface drivers and timing.
- Local two-port buffer memory which is shared by the host and the WD2840.
- Arbitration logic to fairly share the buffer memory, especially when both the host and WD2840 demand access at the same time.
- WD2840 Token Access Controller and associated timing
- Media interface consisting of a manchester encoder/decoder and liner drivers
- Generalized initialization flowchart



**Figure 1. Versabus Application of WD2840 Using Dual Port Memory**

DATA DRIVER ENABLES

NENEN	REC FM. HOST
0 0	ILLEGAL
0 1	OFF
1 0	DRIVE HOST
1 1	

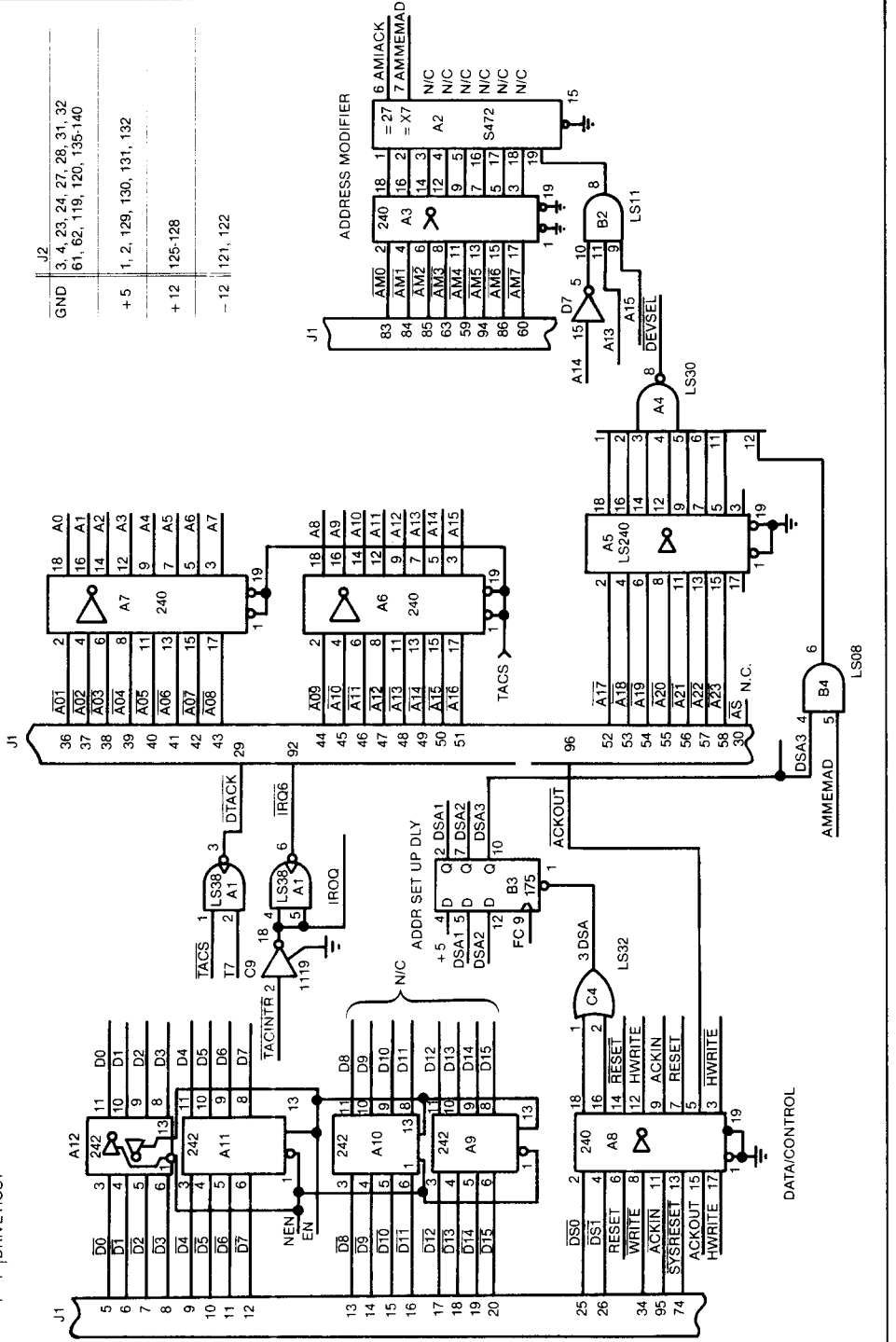


Figure 2. Host Interface

**HOST INTERFACE**

The host interface (Figure 2) utilizes common three-state bus drivers buffering the Versabus from the internal data bus. They are enabled low-Z onto the host bus whenever the host "reads" from the local memory (or WD2840 registers) and are enabled to drive the internal bus whenever the host writes to the internal memory (or registers). All other times these are hi-Z allowing other modules on the Versabus side to use that bus, as well as allowing the WD2840 to use the internal bus. Only an eight bit internal data bus is used mapping all host memory accesses into the lower byte. (Bus drivers A10 and A11 are not used)

Sixteen address bits are buffered and driven onto the internal address bus when the host has access to the RAMs (otherwise the WD2840 drives the internal address bus). The additional address lines of the Versabus are "anded" with the I/O Data Strobe signals DS0 and DS1 and address modifier bits creating a device select signal (DEVSEL) when all are active simultaneously indicating the host actually wants to access this module.

The Host Interrupt and Driver control logic (Figure 3) supplies the host interrupt vector (0FFH) when acknowledged (C10). Acknowledgement occurs when the Versabus ACKIN is received in conjunction with the proper priority level (set at 4 in this design), the proper address modifier (AMIACK), a short de-skew delay (DSA3), and a signal indicating an

interrupt was indeed generated by this module (IROQ).

The logic on this sheet also controls the direction of the data buffers previously described (with signals EN and NEN), presuming the host has active control of the local bus (HOST = 1). The host requests control of the bus for access to the on-board RAM and during interrupt acknowledgement.

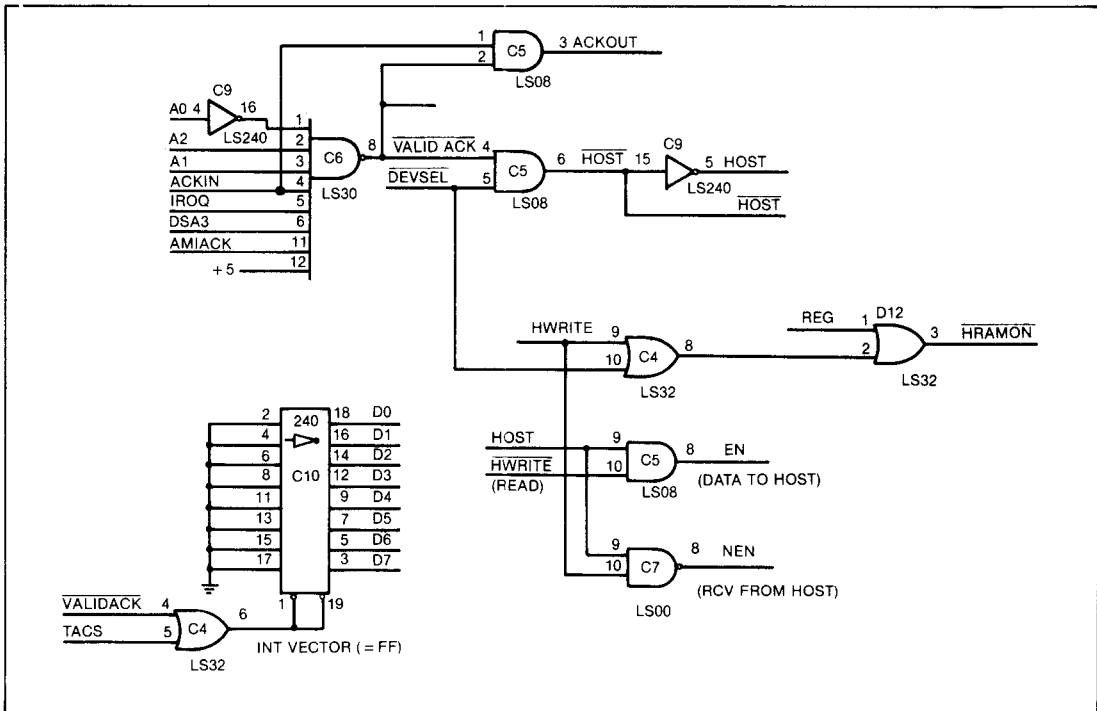
**CLOCK/ARBITRATION**

This logic (Figure 4) generates the synchronous timing used in the rest of the sections. A 16MHz signal derived from a crystal oscillator (part of the manchester logic, described later) is buffered (by C9 and then called FC). This high speed clock is also divided down for the WD2840 system clock CLK at 2 MHz (other slower rates are not used in this design).

This high speed clock clocks a simple latch (B7) until either the host or the WD2840 request local bus access. If the host desires access to this module, HOST is made true, the on-board WD2840 DMA request generated TACDRQ. When either (or both) of these signals occur, IDLE goes false (B5 pin4) freezing the state of latch B7.

IDLE going false starts the timing chain (B1, C1) that generates general timing pulses used later.

When the local memory sequence is complete, at time T10 for the TAC (B8 pin 6) or at time T7 for the



**Figure 3. Host Interrupt and Driver Control**

host (B2 pin 6) a special end of cycle delay is initiated (via shift register E7). This delay ensures that at least 500ns is maintained between WD2840 DMAs and possible host I/O accesses. At the end of this delay, flip-flop B6 generates a one clock "DONE" pulse resetting the arbitration logic.

### MEMORY ARRAY

The memory (Figure 5) uses simple static memories configured as 8K by 8 bits. The RAM data lines are buffered onto the local data bus due to loading considerations. The RAM array is enabled during all I/O operations except those to the first sixteen locations, which are used for accessing the sixteen internal WD2840 registers (REG).

### WD2840 SUPPORT

The WD2840 interface logic is given in Figure 6. The system clock (CLK) is derived from the timing generator (Figure 4). (This clock may be asynchronous with the transmit and receive data clocks if desired.) Address latches are used in this design to provide additional signal drive and to improve memory access timing (the WD2840 does have internal address latches that are useful in less stringent applications).

Host Write (HWRITE) is used to control the direction of I/O operations with the WD2840. When true, the WD2840 expects its internal registers to be written into. This occurs when both WE (pin 3) and CS (pin 4) are both low. Gate C7 1,2,3 ensures that the WE\* signal goes false prior to the data changing (ensures hold time). Chip select logic (D10, 1,2,13,12) enables reads or writes only when the host has access to the internal bus, the internal address bus holds a value in the range of 0-15 (REG true), and a short set-up timer has expired (T1).

Gate F10 (11,12,13) "ands" the WD2840s DMA input and output requests and presents them to the arbitration logic described earlier (via TACDRQ). The sense of the WD2840 DMA request (input or output) is latched (with E10). The DMA output signal is delayed for RAM setup (T2) and turned off before the data is removed to meet RAM hold timing (T7) and presented to the RAM control logic to generate the write pulse.

### MANCHESTER ENCODER/DECODER

The manchester encoder/decoder used here is a Harris HD-6409 (Figure 7). This device is ideal for use with the WD2840 in that its "invalid manchester output," that detects missing clocks, etc., can be directly connected to the WD2840's SQ input. A 16 Mhz crystal controls the internal digital phase locked loop used for clock recovery and generated the 16 Mhz master clock (FC) used for general timing in this design.

The "modem" consists of a simple RS-422 balanced driver and receiver. More elaborate media interfaces are possible, including FSK and broadband, depending on speed / distance / number of taps / cost requirements.

### INITIALIZATION

Figure 10 "flow chart" gives a generalized method of initializing a WD2840 based communications subsystem. First the WD2840 internal diagnostic are preferred, followed by loading of station parameters. Next the network is tested for activity and potential duplicate addresses. Finally the WD2840 TXEN is set allowing normal network generation. The Host now simply monitors TX and RX chains to sent/receive network data.

### SUMMARY

This application note details a simple WD2840 subsystem designed around the VERSAbus form factor. The on-board RAM makes removes any DMA/host bus access questions from the system design. A very simple line driver allows a number of these modules to communicate at speeds of 1 Mbps.

Note that this application note is intended for illustration only; simpler and more elaborate interfaces are possible.

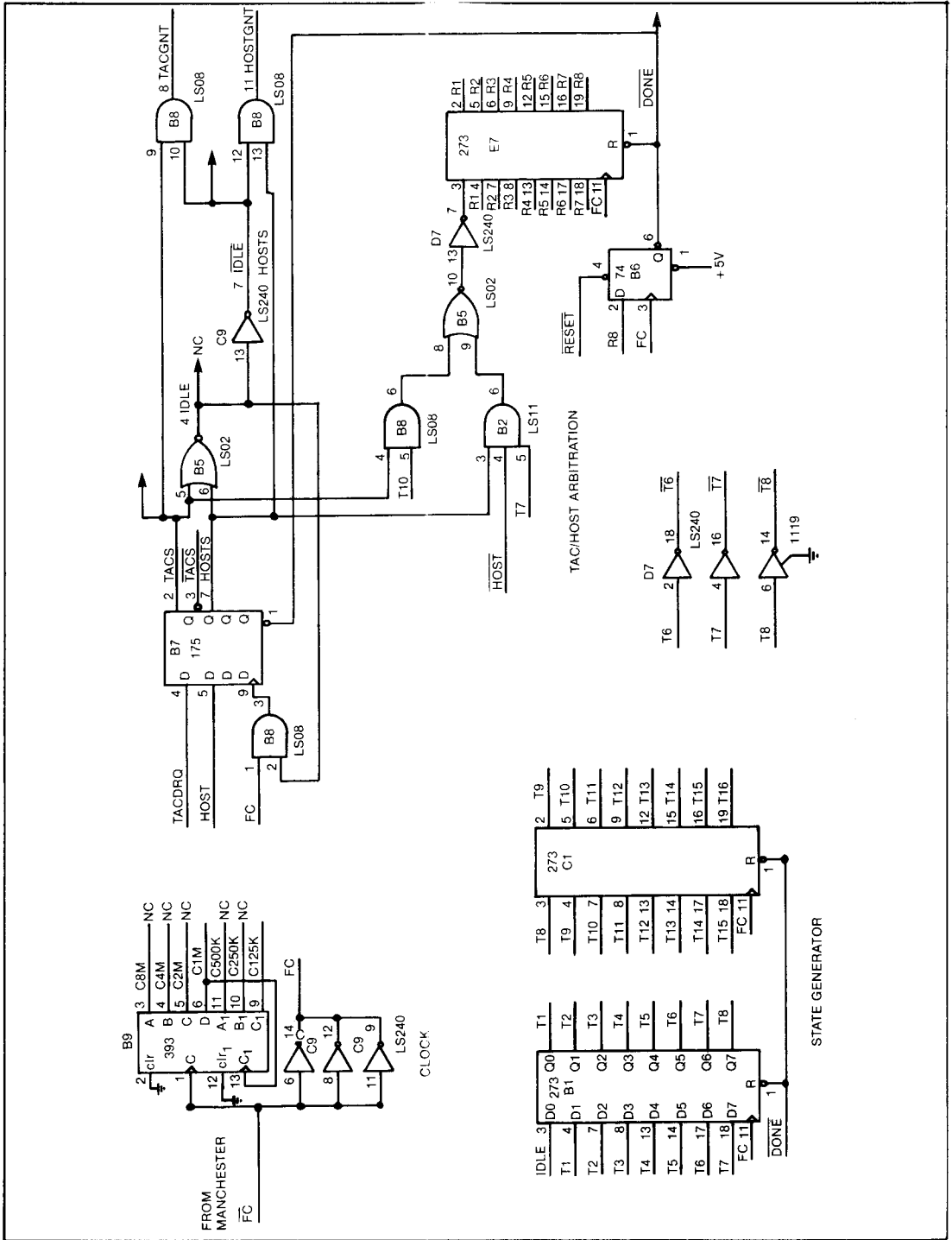


Figure 4. Clock and Arbitration

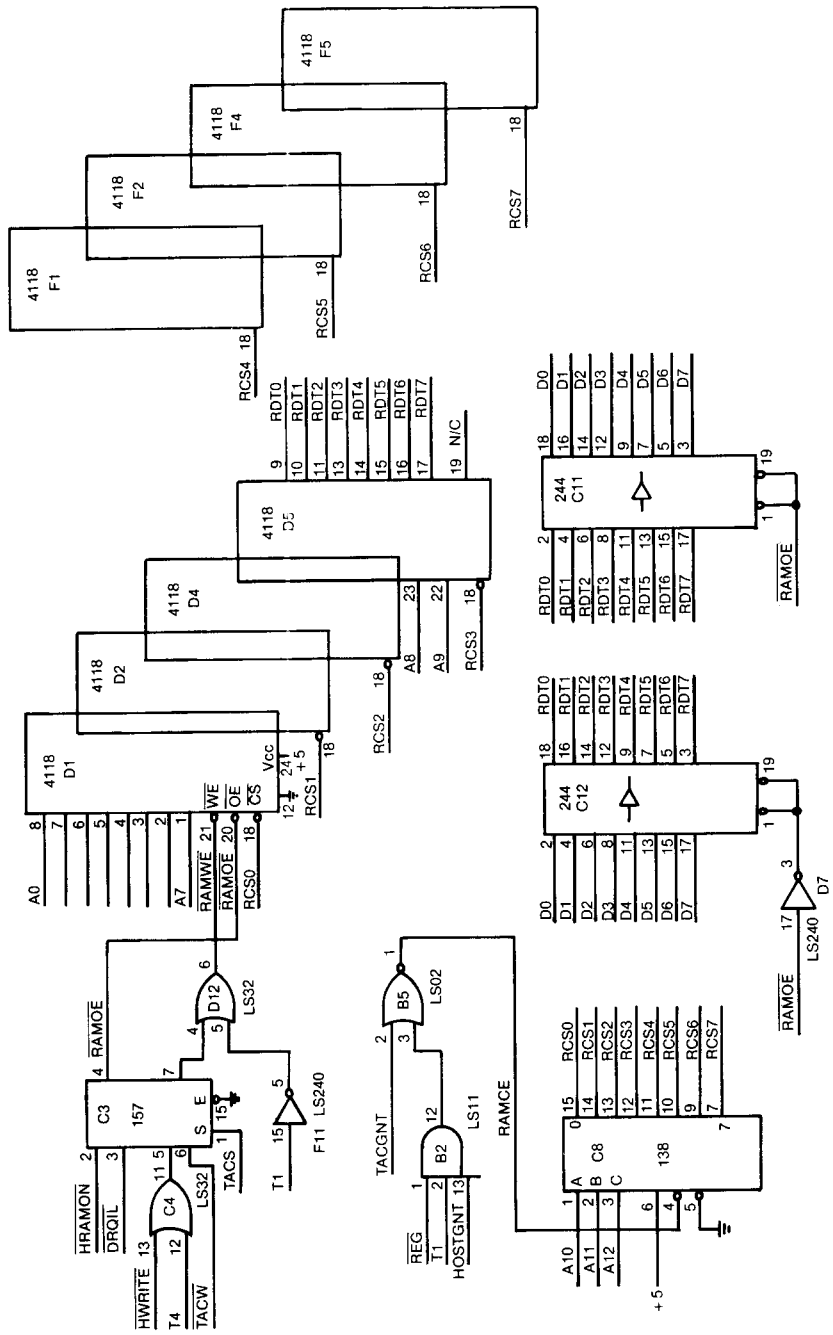
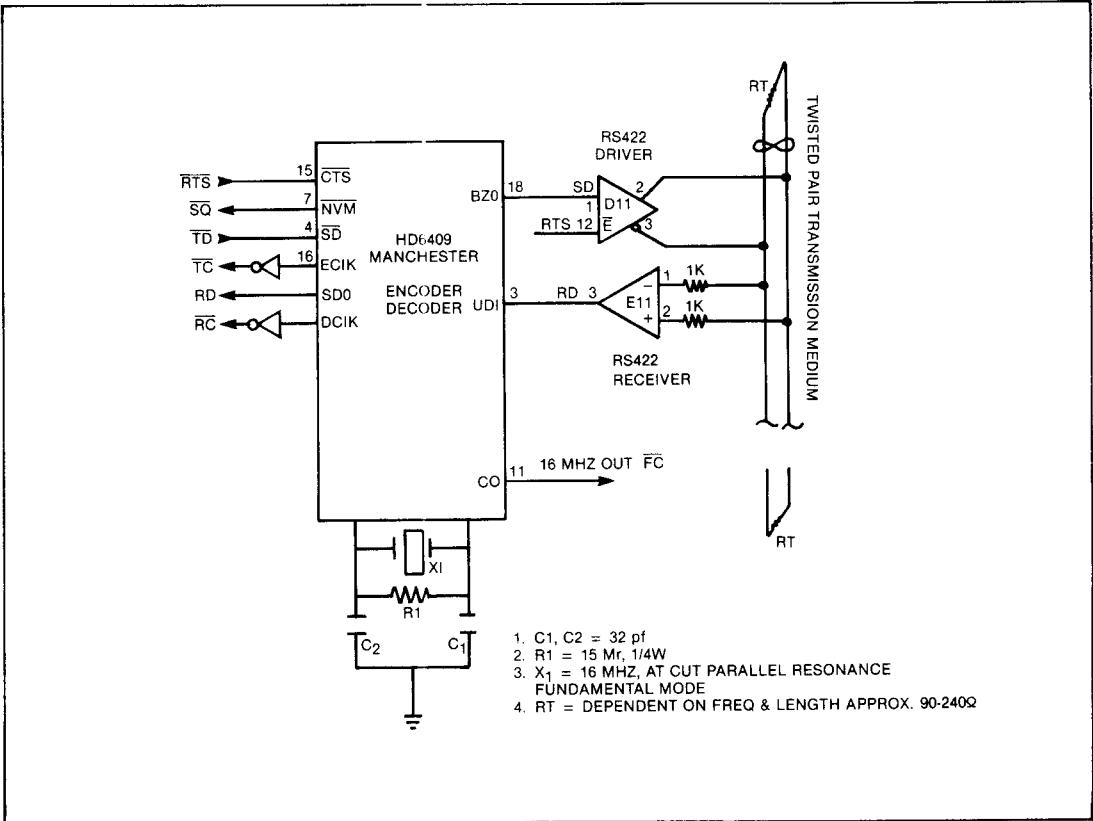


Figure 5. Memory Array





1. C1, C2 = 32 pf
2. R1 = 15 Mr, 1/4W
3. X1 = 16 MHZ, AT CUT PARALLEL RESONANCE FUNDAMENTAL MODE
4. RT = DEPENDENT ON FREQ & LENGTH APPROX. 90-240 $\Omega$

Figure 7. Manchester Encoder/Driver

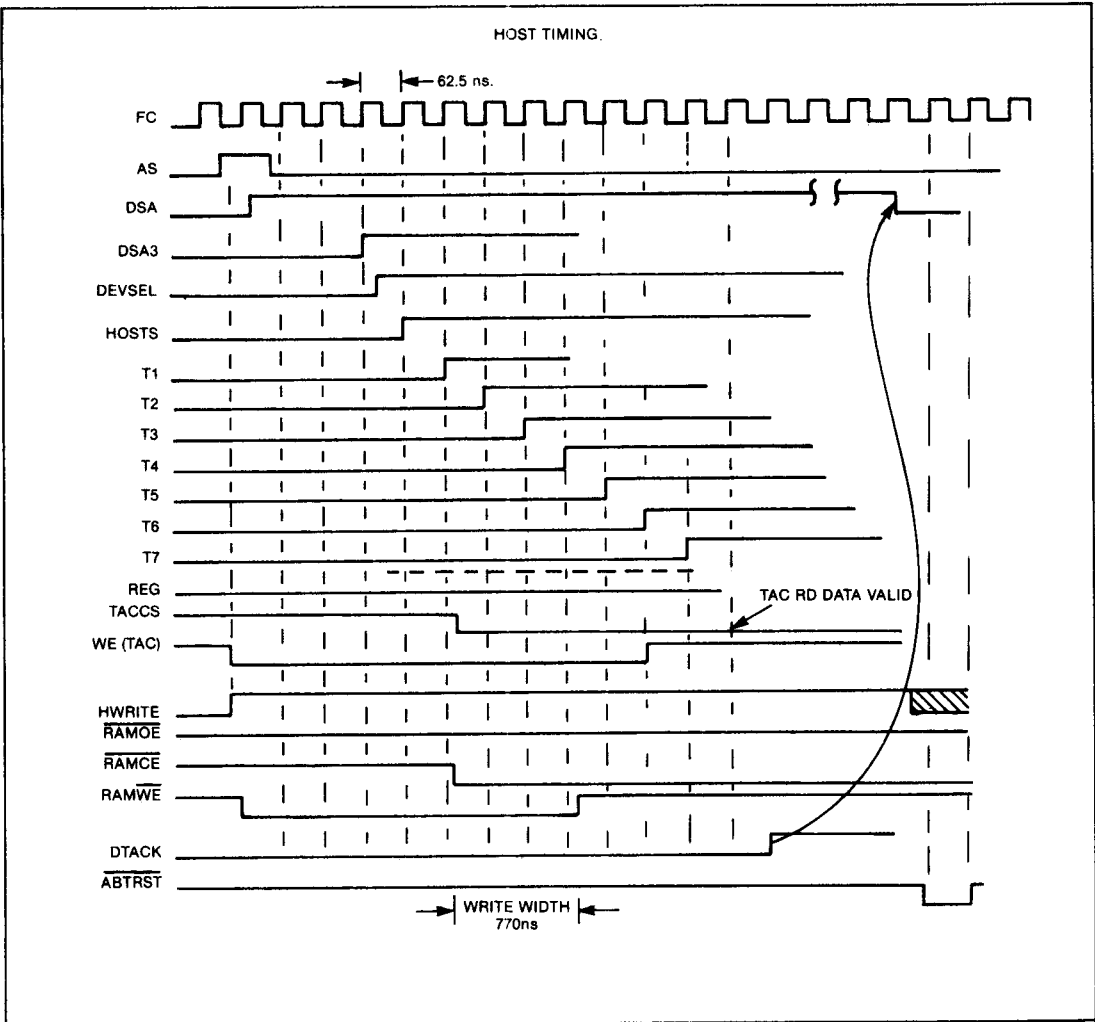


Figure 8. Host Timing

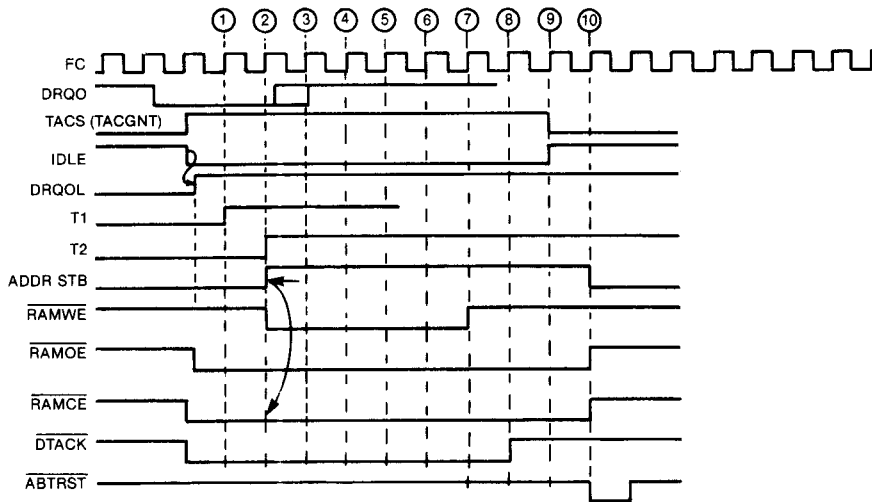


Figure 9. TAC Timing

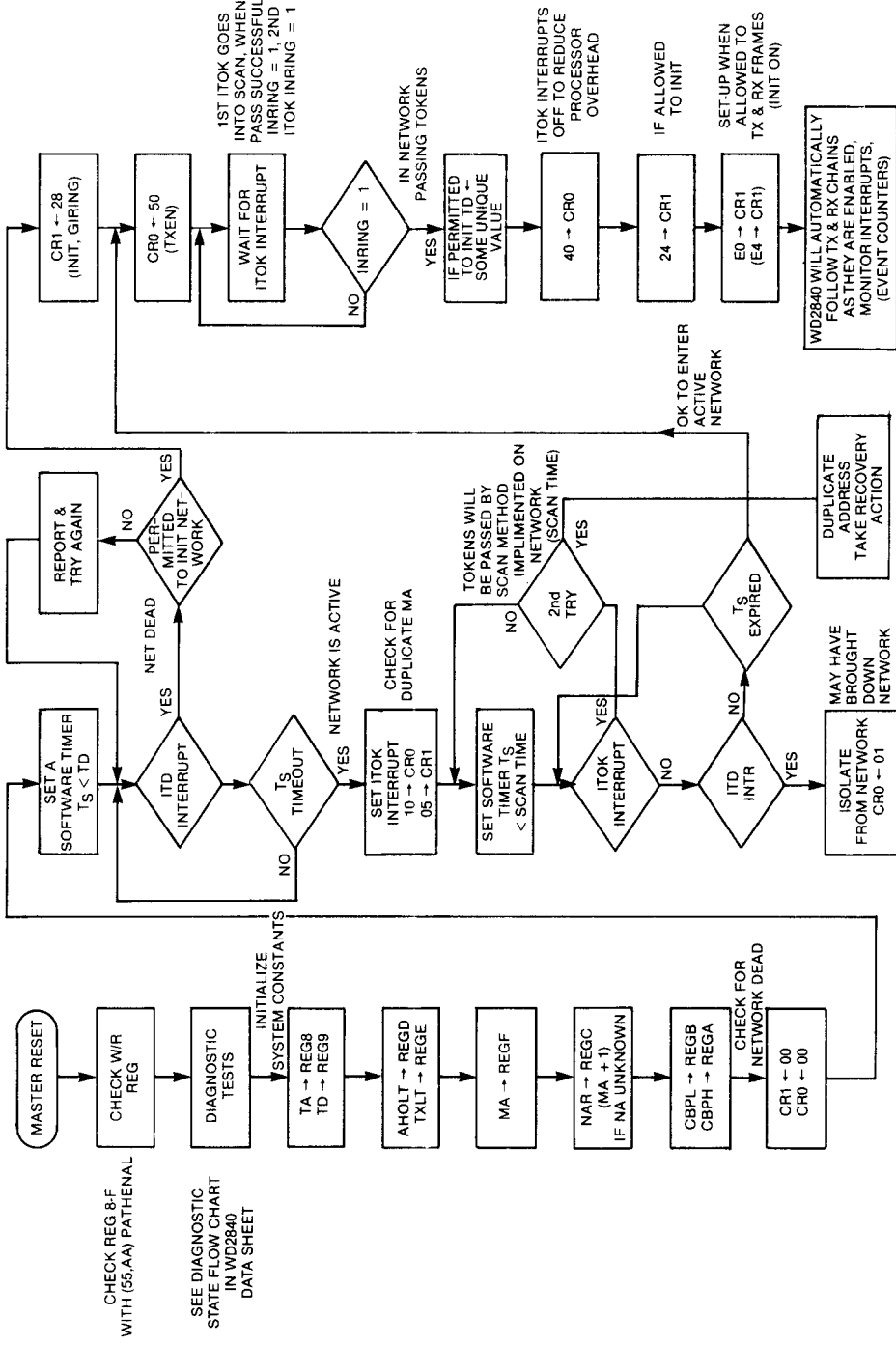


Figure 10. Typical Set-Up Sequence For WD2840