Preliminary

Am79C82A

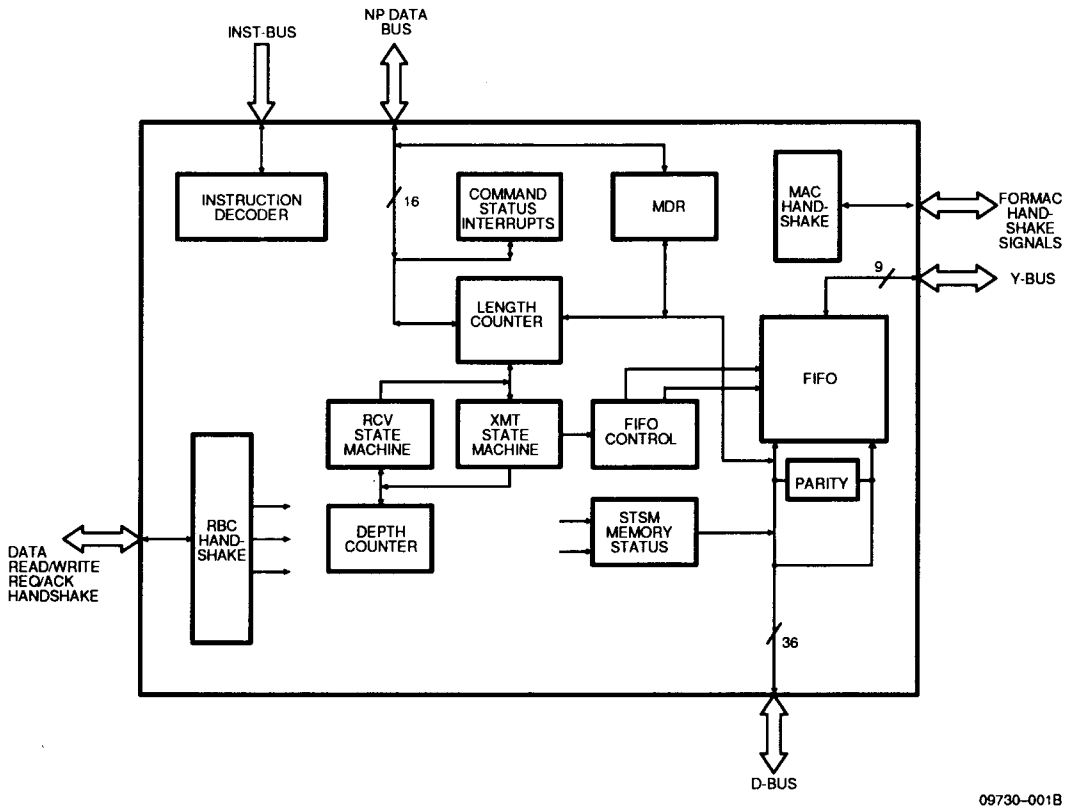
CMOS Data Path Controller (DPC)

Advanced Micro Devices

DISTINCTIVE CHARACTERISTICS

- Performs reception and transmission of frames
- Byte (8 + 1 bits) to word (32 + 4 bits) conversions
- Reports error status
- Performs parity check and generation
- 12.5-MHz byte clock
- 145-lead pin grid array package
- Single +5-V supply

BLOCK DIAGRAM



09730-001B

Publication # 09730 Rev. C Amendment
Issue Date: February 1989

GENERAL DESCRIPTION

The Am79C82A Data Path Controller (DPC) is a CMOS device that, along with the Am79C81A RAM Buffer Controller (RBC), performs buffer management for high-speed interfaces. It buffers and stores the received packets one after the other in Buffer Memory. It also identifies the status of the packet, including any error conditions. This +5-V device runs at 12.5 MHz rate and allows Buffer Memory throughput of 200 Mbps. All inputs and outputs are TTL-compatible, providing a simple interface to other external devices.

NOTE

The word "frame" is used in the SUPERNET data sheets to describe three different groups of information.

1) One group is passed over the network media and has the following structure:

Frame Preamble	Start Delimiter	Frame Control	Destination Address	Source Address	Frame Check	Information	Frame Check Sequence	End Delimiter

2) The others are stored in buffer memory and are structured as follows:
   A) Transmit frame

Descriptor	Frame Control	Destination Address	Source Address	Information	Frame Check Sequence	Pointer

   B) Receive frame

Descriptor	Frame Control	Destination Address	Source Address	Information	Frame Check Sequence

TABLE OF CONTENTS

CONNECTION DIAGRAM
PGA Bottom View (pins facing up)

	A	B	C	D	E	F	G	H	J	K	L	M	N	P	R	
1	NC	VCC	$\overline{CS1}$	$\overline{DS}$	NP14	NP11	NP9	NP8	NP6	NP3	NP1	NP0	$\overline{NMINTR}$	RECEIVE	Y7	1
2	NC	INST0	BMODE	NC	NP15	NP12	NP10	NC	NP7	NP4	NP2	BCLK	VCC	RSHRCVF	Y6	2
3	INST3	INST2	INST1	$\overline{READY}$	GND	NP13	GND	NC	VCC	NP5	GND	$\overline{MINTR}$	DAVALID	Y4	Y5	3
4	NC	DRDREQA	R/$\overline{W}$	●									GND	Y3	NC	4
5	DWRREQ	DRDREQS	$\overline{RESET}$										YP	Y1	Y2	5
6	LDRPXA	BRCVFRM	GND										Y0	MEDREQS	MEDREQA	6
7	VCC	LDRPXS	ERCVFRM										VCC	XMEDAVS	XMEDAVA	7
8	DWRACK	GND	GND				(Bottom View)						RDYTBYT	XFRBYTE	NC	8
9	DRDACKA	DRDACKS	PARERR										GND	XMTABTO	XMTABTI	9
10	DISNHRQ	ACKONE	MDWRACK										INICLBN	RCVABT	MISFRM	10
11	MDRDACK	RBFERR	DP2										GND	NC	TEST	11
12	DP0	DP1	GND										SI	SO	NC	12
13	ODDPAR	DP3	D28	D25	GND	D20	D17	GND	D14	D11	D8	D5	NC	NC	NC	13
14	D31	D30	D27	D24	D22	D19	D16	GND	D13	D10	D7	GND	D3	VCC	NC	14
15	VCC	D29	D26	D23	D21	D18	D15	VCC	D12	D9	D6	D4	D2	D1	D0	15
	A	B	C	D	E	F	G	H	J	K	L	M	N	P	R	

09730-002B

PIN DESIGNATIONS
(Sorted by Pin Number)

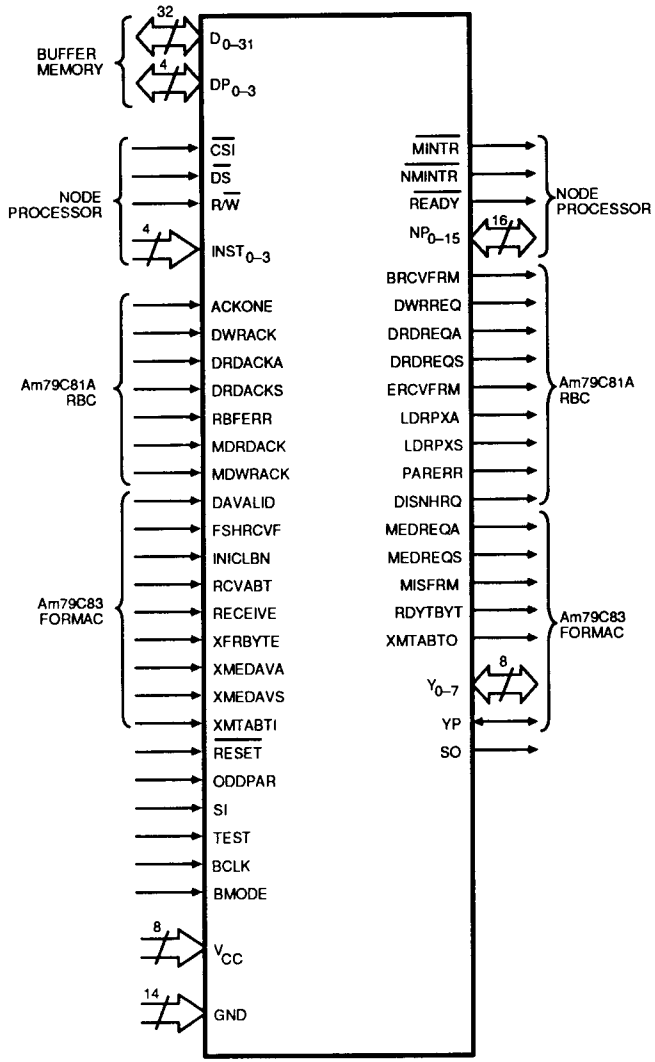
Pin No.	Pin Name	Pin No.	Pin Name	Pin No.	Pin Name	Pin No.	Pin Name
A–1	NC	C–7	ERCVFRM	H–13	GND	N–10	INICLBN
A–2	NC	C–8	GND	H–14	GND	N–11	GND
A–3	$INST_3$	C–9	PARERR	H–15	V_{CC}	N–12	SI
A–4	NC	C–10	MDWRACK	J–1	NP_6	N–13	NC
A–5	DWRREQ	C–11	DP_2	J–2	NP_7	N–14	D_3
A–6	LDRPXA	C–12	GND	J–3	V_{CC}	N–15	D_2
A–7	V_{CC}	C–13	D_{28}	J–13	D_{14}	P–1	RECEIVE
A–8	DWRACK	C–14	D_{27}	J–14	D_{13}	P–2	FSHRCVF
A–9	DRDACKA	C–15	D_{26}	J–15	D_{12}	P–3	Y_4
A–10	DISNHRQ	D–1	$\overline{DS}$	K–1	NP_3	P–4	Y_3
A–11	MDRDACK	D–2	NC	K–2	NP_4	P–5	Y_1
A–12	DP_0	D–3	$\overline{READY}$	K–3	NP_5	P–6	MEDREQS
A–13	ODDPAR	D–4	GUIDE PIN	K–13	D_{11}	P–7	XMEDAVS
A–14	D_{31}	D–13	D_{25}	K–14	D_{10}	P–8	XFRBYTE
A–15	V_{CC}	D–14	D_{24}	K–15	D_9	P–9	XMTABTO
B–1	V_{CC}	D–15	D_{23}	L–1	NP_1	P–10	RCVABT
B–2	$INST_0$	E–1	NP_{14}	L–2	NP_2	P–11	NC
B–3	$INST_2$	E–2	NP_{15}	L–3	GND	P–12	SO
B–4	DRDREQA	E–3	GND	L–13	D_8	P–13	NC
B–5	DRDREQS	E–13	GND	L–14	D_7	P–14	V_{CC}
B–6	BRCVFRM	E–14	D_{22}	L–15	D_6	P–15	D_1
B–7	LDRPXS	E–15	D_{21}	M–1	NP_0	R–1	Y_7
B–8	GND	F–1	NP_{11}	M–2	BCLK	R–2	Y_6
B–9	DRDACKS	F–2	NP_{12}	M–3	$\overline{MINTR}$	R–3	Y_5
B–10	ACKONE	F–3	NP_{13}	M–13	D_5	R–4	NC
B–11	RBFERR	F–13	D_{20}	M–14	GND	R–5	Y_2
B–12	DP_1	F–14	D_{19}	M–15	D_4	R–6	MEDREQA
B–13	DP_3	F–15	D_{18}	N–1	$\overline{NMINTR}$	R–7	XMEDAVA
B–14	D_{30}	G–1	NP_9	N–2	V_{CC}	R–8	NC
B–15	D_{29}	G–2	NP_{10}	N–3	DAVALID	R–9	XMTABTI
C–1	$\overline{CSI}$	G–3	GND	N–4	GND	R–10	MISFRM
C–2	BMODE	G–13	D_{17}	N–5	YP	R–11	TEST
C–3	$INST_1$	G–14	D_{16}	N–6	Y_0	R–12	NC
C–4	$R/\overline{W}$	G–15	D_{15}	N–7	V_{CC}	R–13	NC
C–5	$\overline{RESET}$	H–1	NP_8	N–8	RDYTBYT	R–14	NC
C–6	GND	H–2	NC	N–9	GND	R–15	D_0
		H–3	NC				

PIN DESIGNATIONS

(Sorted by Pin Name)

Pin No.	Pin Name	Pin No.	Pin Name	Pin No.	Pin Name	Pin No.	Pin Name
B–10	ACKONE	A–14	D_{31}	B–7	LDRPXS	A–15	Vcc
M–2	BCLK	N–3	DAVALID	A–11	MDRDACK	A–7	Vcc
C–2	BMODE	A–10	DISNHRQ	C–10	MDWRACK	B–1	Vcc
B–6	BRCVFRM	A–12	DP_0	R–6	MEDREQA	H–15	Vcc
C–1	$\overline{CSI}$	B–12	DP_1	P–6	MEDREQS	J–3	Vcc
R–15	D_0	C–11	DP_2	M–3	$\overline{MINTR}$	N–2	Vcc
P–15	D_1	B–13	DP_3	R–10	MISFRM	N–7	Vcc
N–15	D_2	A–9	DRDACKA	N–1	$\overline{NMINTR}$	P–14	Vcc
N–14	D_3	B–9	DRDACKS	M–1	NP_0	P–8	XFRBYTE
M–15	D_4	B–4	DRDREQA	L–1	NP_1	R–7	XMEDAVA
M–13	D_5	B–5	DRDREQS	L–2	NP_2	P–7	XMEDAVS
L–15	D_6	D–1	$\overline{DS}$	K–1	NP_3	R–9	XMTABTI
L–14	D_7	A–8	DWRACK	K–2	NP_4	P–9	XMTABTO
L–13	D_8	A–5	DWRREQ	K–3	NP_5	N–6	Y_0
K–15	D_9	C–7	ERCVFRM	J–1	NP_6	P–5	Y_1
K–14	D_{10}	P–2	FSHRCVF	J–2	NP_7	R–5	Y_2
K–13	D_{11}	B–8	GND	H–1	NP_8	P–4	Y_3
J–15	D_{12}	C–12	GND	G–1	NP_9	P–3	Y_4
J–14	D_{13}	C–6	GND	G–2	NP_{10}	R–3	Y_5
J–13	D_{14}	C–8	GND	F–1	NP_{11}	R–2	Y_6
G–15	D_{15}	E–13	GND	F–2	NP_{12}	R–1	Y_7
G–14	D_{16}	E–3	GND	F–3	NP_{13}	N–5	YP
G–13	D_{17}	G–3	GND	E–1	NP_{14}	A–1	NC
F–15	D_{18}	H–13	GND	E–2	NP_{15}	A–2	NC
F–14	D_{19}	H–14	GND	A–13	ODDPAR	A–4	NC
F–13	D_{20}	L–3	GND	C–9	PARERR	D–2	NC
E–15	D_{21}	M–14	GND	C–4	$R/\overline{W}$	H–2	NC
E–14	D_{22}	N–11	GND	B–11	RBFERR	H–3	NC
D–15	D_{23}	N–4	GND	P–10	RCVABT	N–13	NC
D–14	D_{24}	N–9	GND	N–8	RDYTBYT	P–11	NC
D–13	D_{25}	D–4	GUIDE PIN	D–3	$\overline{READY}$	P–13	NC
C–15	D_{26}	N–10	INICLBN	P–1	RECEIVE	R–12	NC
C–14	D_{27}	B–2	$INST_0$	C–5	$\overline{RESET}$	R–13	NC
C–13	D_{28}	C–3	$INST_1$	N–12	SI	R–14	NC
B–15	D_{29}	B–3	$INST_2$	P–12	SO	R–4	NC
B–14	D_{30}	A–3	$INST_3$	R–11	TEST	R–8	NC
		A–6	LDRPXA				

LOGIC SYMBOL

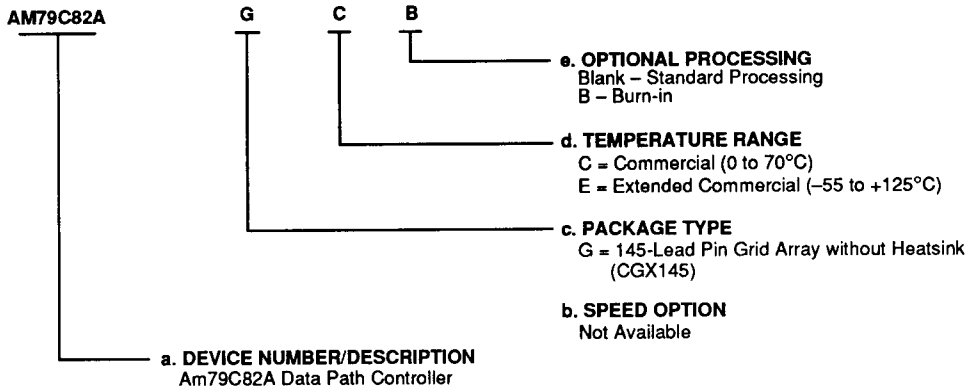


09730-003B

ORDERING INFORMATION
Standard Products

AMD standard products are available in several packages and operating ranges. The ordering number (Valid Combination) is formed by a combination of:

a. Device Number
b. Speed Option (if applicable)
c. Package Type
d. Temperature Range
e. Optional Processing



Valid Combinations	
AM79C82A	GC, GCB, GE, GEB

Valid Combinations

Valid Combinations list configurations planned to be supported in volume for this device. Consult the local AMD sales office to confirm availability of specific valid combinations, to check on newly released combinations, and to obtain additional data on AMD's standard military grade products.

Buffer Memory Interface

D_{0-31}
Data Bus (Input/Output; Three State)

The D-bus interfaces the DPC to the Buffer Memory or any external logic. These lines are outputs whenever the DPC writes frame data, status and length, or a zero field into the memory. When not actively driving out, these pins are always in the input mode. The DPC uses input data from the Buffer Memory for frame transmission when DRDACK is HIGH.

DP_{0-3}
D-bus Parity (Input/Output; Three State)

The DP-bus contains the parity bits for the D-bus. DP_0 contains parity for D_{0-7}, DP_1 for D_{8-15}, DP_2 for D_{16-23}, and DP_3 for D_{24-31}. The parity can be either even or odd based on the state of the ODDPAR pin. No parity provision is provided for the data on the NP-bus. When the MSYPAR bit in the mode register is LOW, then no parity is used in the system and the DP lines are never outputs.

When MSYPAR is HIGH, then the FORMAC, the Host, the NP, and the Buffer Memory have parity. In this case, the DP lines are outputs whenever a received frame, its status and length, or a zero field are being written. Internally the parity logic just checks the parity when the frame is stored; it generates parity for status and length and the zero field. Otherwise these lines are inputs, and the parity is just checked for correctness when either an NP or Host read or write request is serviced.

Node Processor (NP) Interface

BMODE
Bus Mode (Input; Active HIGH)

BMODE indicates that the NP is operating synchronously with the DPC. In asynchronous operation, BMODE is LOW. BMODE should be used as a strapping pin. This pin only affects the handshake on the NP-bus, $\overline{DS}$, $\overline{CSI}$, and $\overline{READY}$ lines.

$\overline{CSI}$
Chip Select Input (Input; Active LOW)

The $\overline{CSI}$ LOW indicates that the NP has selected the DPC to execute an instruction. Internal execution of instructions is only enabled if $\overline{CSI}$ and $\overline{DS}$ are both LOW. $\overline{CSI}$ and $\overline{DS}$ are ANDed internal to the RBC.

When $\overline{CSI}$ is HIGH, the $INST_{0-3}$ and R/$\overline{W}$ lines are ignored and the NP-bus is kept in the three-state mode.

$\overline{DS}$
Data Strobe (Input; Active LOW)

The $\overline{DS}$ signal, together with $\overline{CSI}$, acts as an instruction enable when the NP is synchronous to the BCLK as specified by BMODE pin being HIGH. $\overline{CSI}$ and $\overline{DS}$ are ANDed internal to the RBC.

$\overline{DS}$ is used for defining the presence of data on the NP-bus when the NP is asynchronous to the BCLK (as specified by BMODE being LOW). In this case, $\overline{DS}$ behaves much like the data strobes on slave devices connected to standard microprocessors. When R/$\overline{W}$ and $\overline{CSI}$ are LOW, a write operation is in progress from the NP to the DPC on the NP-bus, and $\overline{DS}$ should become LOW whenever the data to be written is valid. It should stay active and the data on the NP-bus should stay valid until the DPC takes the $\overline{READY}$ line LOW.

In case of a read operation when R/$\overline{W}$ is HIGH and $\overline{CSI}$ is LOW, the NP makes $\overline{DS}$ go LOW whenever it is ready to accept the read data. Subsequently, the DPC provides valid data and also asserts $\overline{READY}$ LOW. After this, the NP can take $\overline{DS}$ HIGH any time it wants. Until it does so, the DPC provides valid data on the NP-bus as output.

$INST_{0-3}$
Instruction Lines (Inputs)

The four instruction lines provide a series of instructions to the DPC. These instructions are valid whenever $\overline{DS}$ and $\overline{CSI}$ are both LOW.

When the NP runs on a clock that is synchronous to the BCLK, as indicated by the BMODE pin being HIGH, the $INST_{0-3}$ lines, R/$\overline{W}$, $\overline{DS}$, and $\overline{CSI}$ should be active for an integral number of BCLK cycles.

When the NP runs asynchronous to the BCLK, as indicated by the BMODE pin being LOW, the $INST_{0-3}$ lines, R/$\overline{W}$, $\overline{DS}$, and $\overline{CSI}$ should be active until $\overline{READY}$ goes LOW. A LOW state on the R/$\overline{W}$ pin indicates a write into the DPC and a HIGH indicates a read from the DPC on the NP-bus.

$\overline{MINTR}$
Maskable Interrupt (Output; Open Drain, Active LOW)

The $\overline{MINTR}$ signal is used to interrupt the NP. This signal goes LOW one BCLK cycle after an unmasked status bit gets set because of an interrupt condition. The maskable bits in the status register of the DPC may be masked out to prevent generation of this interrupt. $\overline{MINTR}$ stays LOW until the status register is read, at which time it goes HIGH. If a status bit is set during the same cycle that the status register is read, then $\overline{MINTR}$ will return LOW on the following cycle. $\overline{MINTR}$ can thus be used with either edge- or level-sensing interrupt controllers.

$\overline{NMINTR}$
Non-Maskable Interrupts (Output; Open Drain, Active LOW)

This signal is similar to the $\overline{MINTR}$ signal except that the status bits that cause $\overline{NMINTR}$ to go LOW cannot be masked out by programming. If any of these selected

status bits are set, then $\overline{\text{NMINTR}}$ will go LOW on the following BCLK cycle and will stay LOW until the NP reads the status register. The status bits are then cleared unless the bit is being set at the same time that the status register is being read. Following a status register read, $\overline{\text{NMINTR}}$ will return HIGH. If another status bit was set on the same cycle as the status register was read, then $\overline{\text{NMINTR}}$ will return LOW on the following BCLK cycle.

NP_{0-15}
Node Processor Bus (Input/Output; Three State)

The NP-bus carries the data for initializing and reading or writing the various registers in the DPC. The use of this bus by the NP for reading or writing does not depend on the activity on the Y-bus or the D-bus.

The NP-bus lines are outputs whenever a read instruction is provided to the DPC by the NP using the instruction lines. This condition is signified by R/$\overline{\text{W}}$, $\overline{\text{CSI}}$, and $\overline{\text{DS}}$ being active. The NP-bus is three-stated if the $\overline{\text{DS}}$ or $\overline{\text{CSI}}$ lines become inactive. Note that the instruction lines, $\overline{\text{CSI}}$ and $\overline{\text{DS}}$ can all be asynchronous with respect to the BCLK, and hence the duration the NP-bus as an output is independent of BCLK when the BMODE pin is LOW.

$\overline{\text{READY}}$
Ready Line (Output; Open Drain, Active LOW)

The $\overline{\text{READY}}$ signal is a handshake signal for use with an asynchronous NP. When the BMODE pin is HIGH, $\overline{\text{READY}}$ is also HIGH and is typically not used in the system. When BMODE is LOW, the NP runs asynchronous to the BCLK; $\overline{\text{READY}}$ indicates when the instruction is acted upon. In case of a write instruction, $\overline{\text{READY}}$ goes LOW after the data is clocked into the DPC. In case of a read instruction, $\overline{\text{READY}}$ goes LOW after the DPC supplies valid data on the NP-bus.

R/$\overline{\text{W}}$
Read/$\overline{\text{Write}}$ (Input)

When R/$\overline{\text{W}}$ is LOW, data is written into one of the RBC's registers or an instruction is executed. When R/$\overline{\text{W}}$ is HIGH, data is read from an RBC register onto the NP-bus. R/$\overline{\text{W}}$ should be controlled in the same manner that $INST_{0-3}$ are.

RAM Buffer Controller (RBC) Interface

ACKONE
Acknowledge First Cycle (Input; Active HIGH)

The ACKONE signal, generated by the RBC, indicates that the current clock cycle is the first half of a two-cycle memory transfer. The DPC uses this in conjunction with the DWRACK signal to generate the proper timing to drive the D_{0-31} and DP_{0-3} pins.

BRCVFRM
Beginning of Received Frame (Output; Active HIGH)

BRCVFRM indicates to the RBC that the following DWRREQ is a request for the first long word of the received frame. BRCVFRM is asserted for one clock cycle and is synchronous with BCLK. BRCVFRM becomes HIGH one cycle after the DPC begins receiving a frame (RECEIVE and DAVALID become HIGH).

DISNHRQ
Disable Node, Host Processor Request (Output; Active HIGH)

DISNHRQ is asserted by the DPC to indicate to the RBC that the DPC requests a continuous access of memory. DISNHRQ becomes HIGH to guarantee that the DPC will get an acknowledgment within one BCLK after it requested to read from or to write into memory. DISNHRQ is given to avoid the overflow of internal FIFO, to keep inter-frame gap short between two frames during reception, and to improve speed of transmission.

DISNHRQ is a level signal and is synchronous with BCLK. During receive, DISNHRQ becomes HIGH at the beginning of a receiving frame, and also at the end of the receiving frame to allow the DPC to write status-length and zero field into memory. During transmit, DISNHRQ can become HIGH for various reasons, such as when the DPC processes a descriptor or a pointer to locate the next frame in a chained transmit mode.

DRDACKA
DPC Read Acknowledge A-frame (Input; Active HIGH)

DRDACKA is an RBC handshake response to the DPC read request DRDREQA. DRDACKA going HIGH allows the DPC to read a long word from the D-bus into the DPC internal FIFO. The RBC will only generate DRDACKA in response to DRDREQA.

DRDACKS
DPC Read Acknowledgment S-frame (Input; Active HIGH)

The DRDACKS is an RBC handshake response to the DPC read request DRDREQS. Each DRDACKS signal is synchronous with BCLK and lasts for two BCLKs. The RBC generates this acknowledge in response to DRDREQS within two BCLK cycles since this request gets highest priority. When DRDACKS becomes HIGH, the DPC assumes that a 32-bit long word is ready on the D-bus. The DPC takes DRDREQS LOW and receives data from the D-bus into its internal FIFO. The RBC will

not generate DRDACKS unless the DPC requests a read of memory for transmitting an S-frame, using DRDREQS.

DRDREQA

DPC Read Request For A-frame (Output; Active HIGH)

The DRDREQA signal is asserted by the DPC to request from the RBC a DMA address for reading a long word of an A-frame into the DPC internal FIFO. DRDREQA goes HIGH only when the DPC transmits A-frames, the MEDREQA and XMEDAVA are HIGH, and when the DPC needs to fill up its internal FIFO with A-frame data to guarantee continuity of frames transmitted. The DPC guarantees that DRDREQS, DRDREQA, and DWRREQ are never HIGH simultaneously.

DRDREQS

DPC Read Request For S-frame (Output; Active HIGH)

The DRDREQS signal is asserted by the DPC to the RBC to request a DMA address for reading a long word (32-bit D-bus) of S-frame data from memory into the DPC internal FIFO. The RBC responds to this by asserting DRDACKS whenever its address output points to the right memory location. The DRDREQS is a level signal synchronous to BCLK. It goes HIGH only whenever the DPC transmits S-frames (MEDREQS and XMEDAVS are HIGH), and when the DPC needs to fill up its internal FIFO with S-frame data to guarantee continuity of frames transmitted.

DRDREQS will not become HIGH when either DRDREQA or DWRREQ are HIGH.

DWRACK

DPC Write Acknowledge (Input; Active HIGH)

DWRACK is the RBC handshake response to the DPC write request DWRREQ going HIGH. DWRACK is a signal synchronous with BCLK, and lasts for multiples of two clock cycles. When DWRACK goes HIGH, the DPC assumes that the address on the RBC address bus is valid. The DPC enables data from its FIFO onto the D-bus. For each DWRREQ, one DWRACK is generated and lasts for two clock cycles. The RBC should assert DWRACK within two clock cycles after DWRREQ becomes HIGH. The RBC will usually not generate DWRACK unless DWRREQ is HIGH. The only exception to this rule is when the DPC writes the status and length of a received frame into memory. In this case, DWRACK is generated in response to ERCVFRM and remains HIGH for four clock cycles.

DWRREQ

DPC Write Request (Output; Active HIGH)

DWRREQ is a signal from the DPC to the RBC to request a memory write cycle for the frame being received. The DPC asserts DWRREQ when it is in the receive mode and its FIFO buffer has enough data to write one long word into memory. When the DPC writes status and length of the received frame into memory, it does not assert DWRREQ.

DWRREQ stays HIGH until the RBC asserts DWRACK. However, it will not go HIGH when DRDREQA or DRDREQS is HIGH. The DPC can keep DWRREQ HIGH or assert another DWRREQ after the previous one is serviced.

ERCVFRM

End of Received Frame (Output; Active HIGH)

The ERCVFRM signal indicates to the RBC that the received frame is ended. ERCVFRM commands the RBC to move the pointer back to the beginning of the frame so that the DPC can write frame status and length into memory. ERCVFRM is asserted one BCLK cycle after the RBC brings DWRACK HIGH to write the last few bytes of the receiving frame in the memory. ERCVFRM is synchronous with BCLK and lasts for one BCLK cycle.

LDRPXA

Load RPX For A-frame (Output; Active HIGH)

LDRPXA instructs the RBC to load data from D_{0-15} to the RPXA pointer. This signal is given to support the chain transmit scheme of the FDDI. This signal goes together with DRDREQA and DRDACKA and is used to load the RPXA pointer. LDRPXS and LDRPXA do not become HIGH simultaneously. LDRPXA is synchronous with the clock. It is asserted simultaneously with DRDREQA and stays asserted until two clock cycles after the DPC receives DRDACKA. LDRPXA is not asserted unless XMEDAVA is HIGH.

LDRPXS

Load RPX For S-frame (Output; Active HIGH)

The primary function of LDRPXS is to instruct the RBC to load data from D_0–D_{15} to the RPXS register (or into the ACP register when in claim or beacon mode). This signal is given to support the chain transmit scheme of the FDDI. LDRPXS is synchronous with BCLK. It is asserted simultaneously with DRDREQS and stays HIGH until two clock cycles after the DPC receives DRDACKS. In primary operation, LDRPXS is not asserted unless XMEDAVS is HIGH.

The secondary function of LDRPXS is to signal the RBC that the DPC's XMTABTI signal has been asserted. In this function, LDRPXS is asserted for one clock cycle on the cycle following the assertion of the XMTABTI signal.

MDRDACK

Memory Data Register (MDR) Read Acknowledgement (Input; Active HIGH)

MDRDACK is asserted by the RBC to allow the DPC to read data from the D-bus into its MDRU and MDRL registers. The DPC also checks parity on the D-bus.

MDWRACK

Memory Data Register Write Acknowledgment (Input; Active HIGH)

MDWRACK is asserted by the RBC to allow the DPC to load data from the MDRU and MDRL registers onto the D-bus. The DPC also generates parity on the D-bus parity pins (DP_{0-3}) if the MSYPAR bit on the mode register is set.

PARERR

Parity Error (Output; Active HIGH)

PARERR HIGH indicates that the DPC parity checking logic has detected an error. Parity is checked on all read transfers out of the Buffer Memory. Parity is generated when the DPC writes into memory. Parity is also checked during all other writes (including DPC, NP, and Host writes) into memory. PARERR is a pulse signal, synchronous with BCLK. It lasts for one clock cycle. Parity can be even or odd depending on the state of the ODDPAR pin.

RBFERR

Receive Buffer Full Error (Input; Active HIGH)

The RBFERR signal is asserted by the RBC to indicate that the receive memory buffer is full. RBFERR is a level signal and synchronous with BCLK. When RBFERR becomes HIGH, the DPC terminates the received frame (i.e., aborts its reception and, unlike RCVABT, no status, length, or zero is written).

Fiber Optic Ring Media Access Controller (FORMAC) Interface

DAVALID

Data Valid (Input; Active HIGH)

The DAVALID signal indicates that valid information is present on the Y-bus for input. If DAVALID is HIGH when RECEIVE is HIGH, the DPC inputs the data byte. DAVALID is HIGH for an integral number of BCLK cycles. Hence, if the FORMAC sends a frame in small bursts, DAVALID stays HIGH only for those cycles when the data is being sent. DAVALID can also go HIGH when RECEIVE is LOW. In this case, the value on the Y-bus is interpreted as status by the DPC and is strobed into an internal status latch. DAVALID will not go HIGH when the DPC is in the transmit mode (as indicated by XMEDAVS or XMEDAVA being HIGH).

FSHRCVF

Flush a Received Frame (Input; Active HIGH)

If FSHRCVF is HIGH when RECEIVE is HIGH, the RBC flushes the frame that the DPC is receiving. FSHRCVF can be the output of either the Am79C83 FORMAC or of external address detection logic. FSHRCVF should be LOW during receive for the DPC to accept the frame. FSHRCVF should not become HIGH when the DPC is not receiving. The FORMAC is designed so that FSHRCVF cannot go HIGH in the same cycle in which RECEIVE goes HIGH, and it can only go HIGH when RECEIVE is HIGH.

INICLBN

Initialize Claim and Beacon (Input; Active HIGH)

The INICLBN signal is asserted by the FORMAC to indicate that the node is in a claim or beacon mode. In this mode, the DPC resets its transmit state machine and asserts a MEDREQS to the FORMAC. This pin is used to support FDDI claim and beacon functions.

MEDREQA

Media Request For A-frame Transmit (Output; Active HIGH)

The MEDREQA signal is asserted by the DPC to indicate to the FORMAC that A-frame data is waiting to be transmitted. The FORMAC responds by asserting XMEDAVA whenever it gains access to the medium for transmission. The DPC asserts it whenever the NP gives DPC an instruction to transmit an A-frame chain segment (IENMXTA). A second IENXMTA will queue up a second chain segment and will cause the DPC to assert MEDREQA again. This keeps MEDREQA HIGH when the end of the first chain segment is reached. MEDREQA goes LOW if a transmit abort occurs.

The NP can abort any queued-up frames by issuing an IABTXMT instruction. The FORMAC can also abort a frame by de-asserting XMEDAVA during a transmit. The FORMAC will not generate XMEDAVA unless the DPC requests (using MEDREQA) the medium to transmit A-frames. The FORMAC also will not generate XMEDAVS and XMEDAVA simultaneously.

MEDREQS

Media Request for S-frame Transmit (Output; Active HIGH)

The MEDREQS signal is asserted by the DPC to indicate to the FORMAC that S-frame data is waiting to be transmitted. The FORMAC responds by asserting XMEDAVS whenever it gains access to the medium for transmission. The DPC asserts it whenever the NP gives DPC an instruction to transmit an S-frame chain segment (IENMXTS). A second IENXMTS will queue up a second chain segment and cause the DPC to assert MEDREQS again. This keeps MEDREQS HIGH when the end of the first chain segment is reached. MEDREQS goes LOW if a transmit abort occurs.

The NP can abort any queued-up frames by issuing an IABTXMT instruction. The FORMAC can also abort a frame by deasserting XMEDAVS during a transmit. The FORMAC will not generate XMEDAVS unless the DPC requests (using MEDREQS) the medium to transmit an S-frame.

MISFRM

Missed Frame (Output; Active HIGH)

MISFRM is asserted by the DPC to indicate to the FORMAC that the DPC missed an incoming frame. MISFRM is synchronous with the system clock, and is asserted when the DPC cannot enter a received frame into its FIFO. This is caused by the inter-frame gap between two received frames being too short or when receive lock is set by a reset.

RCVABT

Receive Abort (Input; Active HIGH)

The RCVABT signal provides a method for the FORMAC to abort a frame that is being received. RCVABT should be synchronous with BCLK, and can be HIGH for one or more clock cycles. If RCVABT goes HIGH for one or more cycles when a frame is being received, the frame is aborted. Subsequent frames are received normally if RCVABT is LOW at the time. RCVABT cannot go HIGH in the same cycle that RECEIVE goes HIGH, and it can only go HIGH with the receive envelope.

RDYTBYT

Ready to Transmit Byte (Output; Active HIGH)

The RDYTBYT is an indication from the DPC to the FORMAC that it has filled up its internal FIFO in preparation for a frame transmission and can guarantee transmission of continuous bytes. RDYTBYT goes HIGH after the FIFO is filled in response to XMEDAVS for an S-frame or XMEDAVA for an A-frame, and other conditions for filling the FIFO are met. Normally it stays HIGH until the FORMAC reads the complete frame using the XFRBYTE signal. Then it goes LOW, signifying that it has nothing more to transmit. RDYTBYT is a level signal synchronous to BCLK. It can go LOW before the frame transmission is complete in case of an abort. If the abort is initiated by the FORMAC de-asserting XMEDAVS (or XMEDAVA), RDYTBYT (as well as MEDREQ) goes LOW in the following cycle. If the abort is initiated by the NP using an IABTXMT instruction, or by the DPC due to an error condition, the DPC conveys this information to the FORMAC by de-asserting MEDREQS (or MEDREQA) and RDYTBYT. In addition, XMTABTO will go HIGH for one BCLK. RDYTBYT is never HIGH if MEDREQS or MEDREQA is LOW.

RECEIVE

Receive Frame (Input; Active HIGH)

The RECEIVE signal from the FORMAC indicates to the DPC that frame reception has begun. Each frame is bounded by RECEIVE. Hence, when RECEIVE goes HIGH, the DPC prepares itself for the frame coming in on the Y-bus. While RECEIVE is HIGH, DAVALID indicates whether or not the data on the Y-bus is valid. When RECEIVE goes LOW, it indicates reception for that frame is over.

RECEIVE changes state synchronous with the BCLK. After RECEIVE goes LOW, it can go HIGH again after two BCLK cycles. However, for the second frame to be stored properly, RECEIVE should go HIGH again eight or more BCLK cycles after it goes LOW. If it goes HIGH sooner, the second frame may be ignored, because the inter-frame gap time on the network may not be sufficient for the DPC to receive this frame and store the previous one. DAVALID is used for strobing FORMAC-generated status into the DPC when RECEIVE is LOW. Typically, RECEIVE should not be HIGH when the DPC is in the transmit mode (as indicated by XMEDAVS or XMEDAVA being HIGH).

XFRBYTE

Transfer Byte (Input; Active HIGH)

The XFRBYTE signal is driven HIGH by the FORMAC in response to RDYTBYT. XFRBYTE should be synchronous to BCLK, and can be HIGH for one or more cycles at a time. Valid transmit data is impressed on the Y-bus when XFRBYTE is HIGH.

XMEDAVA

Transfer Media Available for A-frame (Input; Active HIGH)

The XMEDAVA is a FORMAC response to the DPC handshake signal MEDREQA. XMEDAVA goes HIGH when the FORMAC receives a token from the media authorizing transmission, and the FORMAC has determined that A-frames should be transmitted. The DPC then proceeds to transfer A-frame data to the FORMAC for transmission. The FORMAC will not assert XMEDAVA when MEDREQA is LOW or XMEDAVS is HIGH. It also prevents XMEDAVA and XMEDAVS going HIGH at the same time.

XMEDAVA can go LOW for a variety of reasons. During a normal A-frame transmission, the XMEDAVA goes LOW after the frame is transferred from the DPC to the FORMAC (as signified by RDYTBYT going LOW). If XMEDAVA goes LOW when RDYTBYT is still HIGH, it constitutes a transmit abort from the FORMAC to the DPC since the availability of the medium for transmitting an A-frame disappeared before the transmission was complete.

XMEDAVS

Transmit Media Available for S-frame (Input; Active HIGH)

The XMEDAVS is a FORMAC response to the DPC handshake signal MEDREQS. XMEDAVS is a level signal and is synchronous with BCLK. XMEDAVS becomes HIGH when the FORMAC receives a token from the media authorizing transmission, and the FORMAC has determined that S-frames should be transmitted. The DPC then proceeds to transfer S-frame data to the FORMAC for transmission.

XMEDAVS can go LOW for a variety of reasons. During a normal S-frame transmission, XMEDAVS goes LOW after the frame is transferred from the DPC to the FORMAC (as signified by RDYTBYT going LOW). If XMEDAVS goes LOW when RDYTBYT is still HIGH, it constitutes a transmit abort from the FORMAC to the DPC since the availability of the medium disappears before the transmission is complete. The FORMAC will not

drive XMEDAVS HIGH when either the XMEDAVA is HIGH or the DPC transmit S-frame request MEDREQS is LOW.

XMTABTI
Transmit Abort In (Input; Active HIGH)

When the XMTABTI signal is HIGH, it indicates to the DPC to abort a transmit frame. XMTABTI also causes LDRPXS to be asserted one cycle after XMTABTI is asserted. This signals the RBC that a transmit abort has occurred. S-frame and A-frame lock is also set.

XMTABTO
Transmit Abort Out (Output; Active HIGH)

When the DPC XMTABTO signal is HIGH, it indicates that the frame being transmitted was aborted for some reason (such as the IABTXMT instruction) which was detected by the DPC. The XMTABTO is not asserted in response to a XMTABTI signal assertion from the FORMAC.

Other conditions which cause XMTABTO to be asserted are:
1) Underflow of FIFO;
2) Parity error on pointer or descriptor;
3) Parity error on frame;
4) XMEDAVA(S) goes LOW prematurely; and
5) Coding error on pointer or descriptor.

Y_{0-7}
Y-bus (Input/Output; Three State)

The Y-bus primarily carries data between the DPC and the FORMAC. It can also carry status information from the FORMAC when receiving. The Y-bus is an output whenever the DPC is transmitting and the FORMAC wants to accept the transmitted data. Hence, the DPC presents the transmit data on the Y-bus whenever the XFRBYTE input signal becomes HIGH.

The DPC uses the input data on the Y-bus only when DAVALID is HIGH. If RECEIVE and DAVALID are both HIGH, then the data on the Y-bus is interpreted as a byte in the frame. If RECEIVE is LOW and DAVALID is HIGH, then the DPC interprets the data on the Y-bus as frame-related status being sent from the FORMAC to the DPC, and stores it in a status latch. If the DPC does not receive the status field following the data bytes, then the DPC will use the previous frame's status byte for writing to memory.

YP
Y-bus Parity (Input/Output; Three State)

The YP signal indicates the parity of the Y-bus. The parity can be even or odd as specified by the ODDPAR pin. If parity is allowed, then the direction of signal flow on this pin will match that of the rest of the Y-bus (specified by the MSYPAR bit in the mode register).

Miscellaneous

BCLK
Byte Clock (Input)

The BCLK is the main clock that runs the DPC. The bytes received by the DPC are aligned with this clock. The DPC, Buffer Memory, and FORMAC run synchronously with BCLK when the BMODE pin is strapped HIGH.

ODDPAR
Odd Parity (Input; Active HIGH)

The ODDPAR is a strappable option. When HIGH, it indicates that odd parity is in use. Odd parity is defined by this example: If byte = 00000000, correct parity bit is a "1". When the ODDPAR pin is LOW, it indicates even parity. Parity is computed on a byte basis.

$\overline{\text{RESET}}$
$\overline{\text{Reset}}$ (Input; Active LOW)

This is an active-LOW hardware reset that initializes the internal state machines and the mode register in the DPC, and places them in a pre-defined state. This signal can be made active asynchronously at any point in the DPC operation and should stay LOW for at least four BCLK cycles. It can go HIGH asynchronously at any time after that.

Typically, this signal is connected to a power-on reset circuit, and possibly to a debounced reset switch. A hardware reset is necessary on startup to ensure that the DPC is in a known state and does not drive any unwanted outputs. Subsequent resets to the DPC can be software resets using the reset instruction provided in the instruction set.

Note: Instructions following a reset must wait for at least two clock cycles to be effective.

SI
Shift Scan Input (Input; Active HIGH)

The SI pin is input when the TEST pin is active. This pin is used to input test patterns to various internal points for incoming testing. It should be strapped LOW for normal DPC operation.

SO
Shift Scan Output (Output)

This pin is used for test purposes. The SO pin contains valid test-point data when TEST is HIGH. During test (test pin is HIGH), the SO pin serially shifts out the internal test points of the DPC.

TEST

Test (Input; Active HIGH)

This pin is used for test purposes and should always be connected to ground for normal operation. The TEST pin is used to force the DPC into a test mode for silicon testing. In this test mode, various internal points can be monitored on the SO pin.

Power Supply

GND

Ground (Inputs)

There are fourteen GND pins. They must all be connected to the power return.

V_{CC}

Power (Inputs)

These are eight V_{CC} pins. They must all be connected to a +5–V ±5% supply.

FUNCTIONAL DESCRIPTION

Functional Overview

Transmit Mode

Before frames are transmitted, they are stored in Buffer Memory in a linked list of blocks that are chained together by pointers. The chain of frames to be transmitted begins with a pointer to the first block in the chain. Each block consists of a descriptor word, one or two more words of data to be transmitted, and a pointer to the next block. The descriptor contains the length of the data field in bytes plus some other information about how the data is stored. The formats of these fields are described in a later section.

The FDDI standard divides messages into two categories: 1) data that must be transmitted within a specified short period of time (S-frames), and 2) data that can tolerate a longer delay before transmission (A- frames). Each category of data is stored in the Buffer Memory in a separate, independent chain FIFO. The frames in each FIFO are chained so that each frame contains the address of the next frame in the chain.

Receive Mode

Once initialized, the DPC is ready to receive frames from the FORMAC. Frames 1 to 64K bytes long can be written into Buffer Memory using the proper request handshake with the RBC. The DPC-generated frame status and length information are normally written into memory at the start of the frame (when the RBC mode register bit MSLNSTM is set). The status bits generated by the FORMAC can also be written along with the DPC status. This scheme allows for latency of the NP in servicing received frames because the status remains stored on a per-frame basis.

All received frames are stored in Buffer Memory using a FIFO managed by the RBC. The RBC issues a "buffer full" error if a write is attempted when the FIFO is full.

Claim/Beacon Frame Handling

The Claim and Beacon frames are specific patterns stored in the Buffer Memory. The DPC transmits these without software intervention whenever the INICLBN pin is asserted.

User Test Mode

When the TEST pin is HIGH, the DPC is forced into a test mode. Registers which are not accessible from the NP-bus can be configured as a shift register and controlled by writing test patterns into the shift scan input pin (SI) and read back on the shift scan output pin (SO).

Overview of User-Accessible Resources

Programmable Resources

Mode Register (MODE)

This register is used to program various modes of operation of the DPC. It permits received frames to be stored starting at any byte boundary within the 32-bit long word, writing either the least-significant byte first or most-significant byte first. It also allows parity checking and generation to be enabled or disabled. This register can be loaded or read using the ILDMODE or IRDMODE instructions.

Status Register (STAT)

The status register provides an indication of various events seen by the DPC. The status bits, if not masked, can generate an interrupt to the NP. They can be read with an IRDSTAT instruction.

To assist in handling the transmission of frames, the DPC can generate interrupts under the following conditions: 1) after transmitting either individual frames or a chain of frames, 2) when all queued frames are transmitted, or 3) when the transmission is aborted for some reason.

When receiving, status is updated after each frame reception. Error conditions are also indicated by status bits. A status bit is set when the inter-frame gap is too small (i.e., the next frame comes within 6 bytes of the last one and the DPC is not ready to handle it). If the NP and the DPC simultaneously try to modify an internal register, that too is flagged. Also, a DPC internal FIFO overflow or underflow, due to some malfunction, also causes a status bit to be set.

Interrupt Mask (IMSK)

Some of the status bits can be masked using the IMSK register bits so that the $\overline{\text{MINTR}}$ signal is not asserted. This register can be loaded or read with the ILDIMSK or IRDIMSK instructions.

Length Counter (LNCN)

The length counter (LNCN) is a 16-bit counter used to count the length of the frames being transmitted and received.

When transmit begins, the value of transmit frame length (located in memory) is loaded into the LNCN. The LNCN is decremented for each byte of data read into the internal DPC FIFO. The frame-length value specifies the number of bytes to be transmitted, starting from the location specified by the RPX of the RBC, and the byte

specified by the RPXM1 and RPXM0 bits in the transmit frame descriptor.

When receive begins, the LNCN resets itself and counts the bytes of the frame which are stored in the memory. At the end of receive frame, the contents of the LNCN are stored into memory at the beginning of the frame after the entire frame reception is complete.

The ability to read or write the LNCN (using the ILDLNCN or IRDLNCN instructions) is provided for diagnostic reasons and need not be used in normal operation.

Memory Data Register (MDRU and MDRL)

The MDR is a 32-bit holding register in the DPC used to buffer data transferred between the NP and the Buffer Memory when RBC instructions are used rather than DMA transfers. The MDR has access to the NP-bus as well as the D-bus. The high-order 16 bits of MDR are referred to as MDRU, and the lower-order 16 bits are referred to as MDRL. The NP can load MDRU and MDRL using the ILDMDRU and ILDMDRL instructions and the NP-bus for data. These can also be read onto the NP-bus using the IRDMDRU and IRDMDRL instructions.

The contents of MDR are written into the Buffer Memory via the D-bus. The NP can read from the MDR after performing a read-memory instruction using the RBC. It can also write data into the MDR and then perform a write-memory instruction using the RBC. Data transfers take place at the location pointed to by MAR in the RBC.

The use of MDR eliminates the need for an external latching register to capture NP data, as long as the NP can tolerate the two-step access of the MDRU and MDRL.

Instruction Set

The NP can issue software instructions to the DPC using the $INST_{0-3}$, R/$\overline{W}$, and NP_{0-15} lines. $INST_{0-3}$ are normally connected to the address bus of the NP.

Frame Descriptors in Buffer Memory

Receive frame status and length are stored in the receive frame descriptor. This 32-bit field is in the first location in Buffer Memory for a received frame.

Frames to be transmitted are stored with a 32-bit transmit frame descriptor and a 32-bit pointer (containing the address of the transmit frame descriptor for the next frame in the transmit queue). The descriptor is stored in the Buffer Memory word just before the data to be transmitted, while the pointer is stored just after the data.

Hardwired Resources

Using the BMODE pin, the DPC can interface with either a synchronous or asynchronous Node Processor. When BMODE is tied HIGH, the NP clock is synchronous with the DPC clock. In this case, the $\overline{DS}$ and $\overline{CSI}$ pins can be connected together and become LOW when the NP issues an instruction to the DPC. When BMODE is tied LOW, the DPC can interface with an asynchronous NP. Here, the $\overline{CSI}$ pin indicates the presence of an instruction from the NP instruction line, the $\overline{DS}$ pin indicates the presence of data on the NP-bus, and the $\overline{READY}$ line signifies that the DPC received the message from the NP.

Block Diagram

External Buses

Three buses are used to interconnect these blocks to the outside world:

1) **NP-bus:** A 16-bit bus used primarily by the Node Processor (NP) for initializing the DPC.

2) **Y-bus:** An 8-bit bus connected to the FORMAC. It is used to receive parallel data from the FORMAC during frame reception, and to transmit parallel data to the FORMAC during transmission. It also has an associated parity bit.

3) **D-bus:** A 32-bit bus that goes to the Buffer Memory. The D-bus is used for writing received data into the Buffer Memory, and for reading data from the Buffer Memory during transmission. There are four optional parity bits that protect the 32-bit bus during frame transmission and reception.

Instruction Decoder and Handshake

The instruction decoder logic decodes the NP instructions. The handshake synchronizes the instructions to the network clock when the NP runs asynchronous with the DPC.

Internal FIFO Buffer and Control

The RBC arbitrates between requests coming from the DPC and several other sources for the use of the memory. In order to guarantee continuity of the frames, the DPC has an internal FIFO to temporarily store small sections of the frames while the RBC is servicing either an NP or Host request. The internal FIFO buffer and control section consists of an internal FIFO 8 bytes deep and a state machine to control the FIFO. This block also performs a conversion between 8-bit and 32-bit parallel data between the Y-bus and D-bus.

Receive and Transmit State Machines

The receive and transmit state machine controls frame reception and transmission. These state machines are also responsible for halting the DPC in case of fatal-error conditions, and for queuing up a transmit frame while a frame reception is taking place. This section also contains a depth counter to count the depth of the FIFO during receive and transmit. None of the buses communicate with this block.

RBC Handshake

Since the RBC arbitrates requests for memory address generation, the DPC must issue a request to the RBC for DMA use. The RBC handshake-block control performs sequential handshakes between the DPC and the RBC so that the RBC can generate addresses for both reading from and writing to the Buffer Memory.

FORMAC Handshake

During transmit, the NP sets up an appropriate pointer in the RBC and gives the DPC instructions to transmit. The DPC tells the FORMAC that frames are waiting to be transmitted. The DPC then waits until the FORMAC has access to the medium. When the FORMAC receives a token, it tells the DPC to start transferring frames from the Buffer Memory for transmission. The FORMAC handshake block performs this logic handshake sequence between DPC and the FORMAC for transmission.

Parity Logic

The DPC checks parity on all data read from or written to the Buffer Memory. (Note: the DPC also checks parity on all Host and NP reads and writes on the D-bus.)

Parity is organized as 1 bit per byte and may be configured as odd or even. However, when used in an FDDI application in conjunction with the FORMAC, the DPC must be configured for odd parity.

Parity is generated when the DPC writes into Buffer Memory.

Locks

The DPC contains three locks (one for receive and two for transmit), used to effectively disable the appropriate function. Once the DPC is reset, the receive lock is activated which takes the DPC off-line for initialization purposes.

The two transmit locks are set by the following conditions:

1) S-frame lock:
   XMTABTI
   XMEDAVS negated prematurely
   Parity error on pointer or descriptor (S-frame queue)
   Coding error on pointer or descriptor (S-frame queue)

2) A-frame lock:
   Similar to the above S-frame condition but is used for the A-frame queue

All locks can be cleared by the ICLRLCK instruction.

NP — DPC Interface

Clock Synchronization

The NP can run either synchronously or asynchronously with respect to the network clock (BCLK). The BMODE pin is strapped HIGH to indicate a synchronous operation, and strapped LOW to indicate an asynchronous operation. The synchronous case can be one of two kinds.

If the NP runs on a faster clock than the DPC, there are two or more integral NP clock cycles during every DPC cycle. In this case, the NP needs to repeat instructions an integral number of times so that the instruction is active (and glitch-free) for at least one BCLK cycle.

When the NP clock is slower than BCLK, instructions have to be active and glitch-free for at least one BCLK cycle. The DPC executes the instruction only once provided $\overline{DS}$ is LOW for only one BCLK cycle when the instructions are valid. If $\overline{DS}$ stays LOW for several cycles, then the instruction will be executed several times.

Note that in the synchronous case, the "AND" of $\overline{CSI}$ and $\overline{DS}$ signals (complements of the $\overline{CSI}$ and $\overline{DS}$ pins, respectively) is treated as an instruction-enable signal. $\overline{CSI}$ can be asserted for multiple cycles to keep the data on the NP-bus valid for a longer duration. In the synchronous case, the $\overline{READY}$ line is LOW since it is not needed.

In the asynchronous case, a handshake convention is used (as shown in Figure 6 of the Am79C81 data sheet). The NP makes $\overline{CSI}$ and $\overline{DS}$ LOW to give the instruction. This is synchronized inside the DPC. This can take from two to three cycles. The DPC executes this instruction once, and at the end of the execution, $\overline{READY}$ is driven LOW. For a load instruction, $\overline{READY}$ is driven LOW after the data is loaded into an internal register or the instruction is executed. For a read instruction, the read data is latched and provided continuously on the NP-bus when $\overline{CSI}$ and $\overline{DS}$ are LOW. When the NP is done with the instruction, it de-asserts $\overline{CSI}$ and/or $\overline{DS}$. This causes $\overline{READY}$ to become HIGH and complete one instruction operation. If the handshake is violated, the results are undefined.

Instruction Set

The instructions for the DPC are shown in Table 1. During loads, the value on NP_{0-15} is loaded into the register; during reads, the value in the addressed register is enabled onto the bidirectional NP_{0-15} bus.

Most of these instructions are very straightforward and need no description. Some others are described below (by instruction number order):

IRESET (00 Hex). The software reset (IRESET) is the software equivalent of the hardware reset pin ($\overline{RESET}$) and it resets the DPC. During the reset operation, all state machines are initialized, status is cleared, interrupt mask (IMSK) is activated, and the mode register is cleared. The clearing of the STAT and setting of the IMSK instructions ensures at reset that the $\overline{MINTR}$ and $\overline{NMINTR}$ lines are HIGH. The reset also clears all transmit queues for pending frames. After a reset, wait for two clock cycles before attempting another instruction.

ILDLNCN (01 Hex). Used to load a 16-bit value into the length counter. In normal operation this function is not necessary since the DPC reads the frame length from the transmit chain in the Buffer Memory. It is used for diagnostics only.

IABTXMT (08 Hex) (Instruction to Abort Transmit). If an IABTXMT is given when a frame transmission is in progress, this transmission is aborted. This is indicated by RDYTBYT, MEDREQS, and MEDREQA going LOW. XMTABTO will also go HIGH for one BCLK. If given before frame transmission is begun, all queued-up transmit chain queues (for A-frames and S-frames) are aborted and both MEDREQS and MEDREQA go LOW. When a transmit is aborted, the SXMTABT status is set. This command cannot be used to abort a Claim or Beacon frame.

IENXMTS (09 Hex) (Instruction to Enable Transmit S-frame). When the IENXMTS instruction is given to the DPC, the DPC asserts the MEDREQS signal going to the FORMAC, and waits for a XMEDAVS to come back. The NP should set up the frames to be transmitted in the Buffer Memory before giving the IENXMTS instruction to the DPC. This group of frames queued up with the IENXMTS instruction is referred to as a chain segment. A chain segment is defined as a chain of frames in which the MORE bit = 1 for all frames except the last one in which MORE = 0. The MORE bit is the most-significant bit of the descriptor word.

In essence, one IENXMTS enables one chain segment. A second IENXMTS given for a second chain segment linked to the first one is also remembered. A third IENXMTS will be remembered only if all frames in the first chain segment have already been transmitted. As a result, only two chain segments can be pending at any time. There is no constraint on the number of frames in each chain segment; however, none of the frames or their linkages can be modified after an IENXMTS is given.

IENXMTA (10 Hex) (Instruction to Enable Transmit A-frames). IENXMTA functions as IENXMTS except IENXMTA makes the DPC assert MEDREQA and start transmitting A-frames when XMEDAVA is asserted by the FORMAC.

The IENXMTS and IENXMTA should be given in such a way that at any time there are at most two of each chain segment types that are unserviced.

ICLRLCK (11 Hex). This instruction is used to clear the transmit and receive locks in conjunction with NP-bus bits 0, 1, and 2. The receive lock is set by a reset (pin or instruction), and is used to disable all receive functions until the device(s) is(are) initialized completely.

The transmit locks (one for S-frames and one for A-frames) are set by any of the following conditions:

1) XMTABTI;
2) Negation of XMEDAVA(s) when transmission is in progress;
3) Parity error on pointer or descriptor; or
4) Coding error on pointer or descriptor.

The NP-bus bits used to clear the three locks are coded as follows:

	NP-bus Bits			
ICLRLCK	**2**	**1**	**0**	**Action**
Active	x	x	1	Clear S-frame lock
Active	x	1	x	Clear A-frame lock
Active	1	x	x	Clear receive lock

IRDSTAT (16 Hex). IRDSTAT enables the value of the 16-bit status register onto the NP-bus (the meaning of each bit is defined in the status section). IRDSTAT also clears the status register. This autoclear obviates the need for a separate status clear instruction and prevents interlock problems. If3 a status bit is being set during this instruction, then the old value is read, but the new value is written into status by overriding the autoclear. Some of these status conditions generate a non-maskable interrupt on the $\overline{NMINTR}$ pin. The unmasked status conditions generate a maskable interrupt on $\overline{MINTR}$. The IRDSTAT is the means by which the NP responds to these interrupts and finds out about DPC status. The autoclear works on all the bits of STAT.

IRDLNCN (17 Hex). IRDLNCN reads the value of a 16-bit length counter (LNCN) onto the NP-bus. This counter is used primarily to count the number of bytes of the frame during receive and transmit. The IRDLNCN instruction can be given one cycle after RECEIVE goes LOW (which signifies end of frame into the DPC), and results in the value of the frame length being read onto the NP-bus. During receive only, the value of the length counter appears on the NP-bus as a complementary signal. This instruction is provided for diagnostics only, and should not be used during normal operation.

All other load/read instructions are self-explanatory register loads.

Table 1. DPC Instruction Set

Instruction Mnemonic	R/$\overline{W}$	$INST_3$	$INST_2$	$INST_1$	(LSB) $INST_0$	NP_{0-15}	Function
Software Reset:							
IRESET	0	0	0	0	0	X X X X	Software Reset
Instructions to Load and Read DPC Registers:							
ILDLNCN	0	0	0	0	1	W W W W	Load LNCN
ILDMODE	0	0	0	1	0	W W W W	Load MODE
ILDIMSK	0	0	0	1	1	W W W W	Load IMSK
ILDMDRL	0	0	1	1	0	W W W W	Load MDRL
ILDMDRU	0	0	1	1	1	W W W W	Load MDRU
IABTXMT	0	1	0	0	0	X X X X	Abort Transmit
IENXMTS	0	1	0	0	1	X X X X	Enable Transmit for S-frames
IENXMTA	0	1	0	1	0	X X X X	EnableTransmit for A-frames
ICLRLCK	0	1	0	1	1	W W W W	Clear Locks (Receive & Transmit)
IRDSTAT	1	0	0	0	0	R R R R	Read STAT
IRDLNCN	1	0	0	0	1	R R R R	Read LNCN
IRDMODE	1	0	0	1	0	R R R R	Read MODE
IRDIMSK	1	0	0	1	1	R R R R	Read IMSK
IRDMDRL	1	0	1	1	0	R R R R	Read MDRL
IRDMDRU	1	0	1	1	1	R R R R	Read MDRU
Reserved Instructions:							
IRSV1	0	0	1	0	0	—	Reserved
IRSV2	0	0	1	0	1	—	Reserved
IRSV3	0	1	1	0	0	—	Reserved
IRSV4	0	1	1	0	1	—	Reserved
IRSV5	0	1	1	1	0	—	Reserved
IRSV6	0	1	1	1	1	—	Reserved
IRSV7	1	0	1	0	0	—	Reserved
IRSV8	1	0	1	0	1	—	Reserved
IRSV9	1	1	0	0	0	—	Reserved
IRSV10	1	1	0	0	1	—	Reserved
IRSV11	1	1	0	1	0	—	Reserved
IRSV12	1	1	0	1	1	—	Reserved
IRSV13	1	1	1	0	0	—	Reserved
IRSV14	1	1	1	0	1	—	Reserved
IRSV15	1	1	1	1	0	—	Reserved
IRSV16	1	1	1	1	1	—	Reserved

Key: W W W W = Write data to DPC register
R R R R = Read data from DPC register
X X X X = Don't Care

Mode Register (MODE)

The DPC has a 4-bit mode register (MODE). The instruction ILDMODE is used to load 4-bit values from the NP-bus into MODE, and IRDMODE allows the contents of MODE to be read onto the NP-bus.

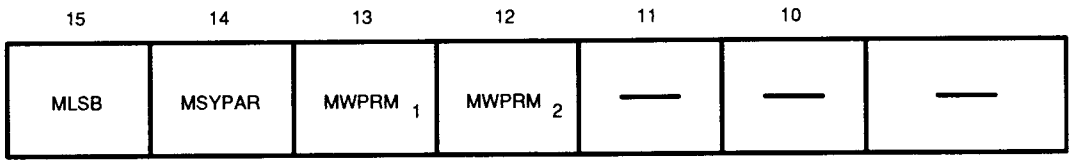
When reset (either through the $\overline{\text{RESET}}$ pin or through software IRESET), MODE is initialized so that all the bits are cleared. The NP then loads the required value into the mode register. MODE bits are shown in Figure 1 and detailed below.

MLSB (Mode Least-Significant Byte). The MLSB HIGH indicates that the least-significant byte in the long word should be transmitted first, as shown in Figure 2. If the MLSB bit is LOW, the most-significant byte is transmitted first; however, the order of bits in a byte is not changed.

MSYPAR (Mode for System Parity Configuration). When LOW, this bit disables parity generation and checking, and the DP_{0-3} and YP lines are three-stated. When MSYPAR is HIGH, parity is enabled. The parity is checked in the DPC at the D-bus for both receive and transmit. For transmit, the Buffer Memory parity is checked and carried through the DPC FIFO.

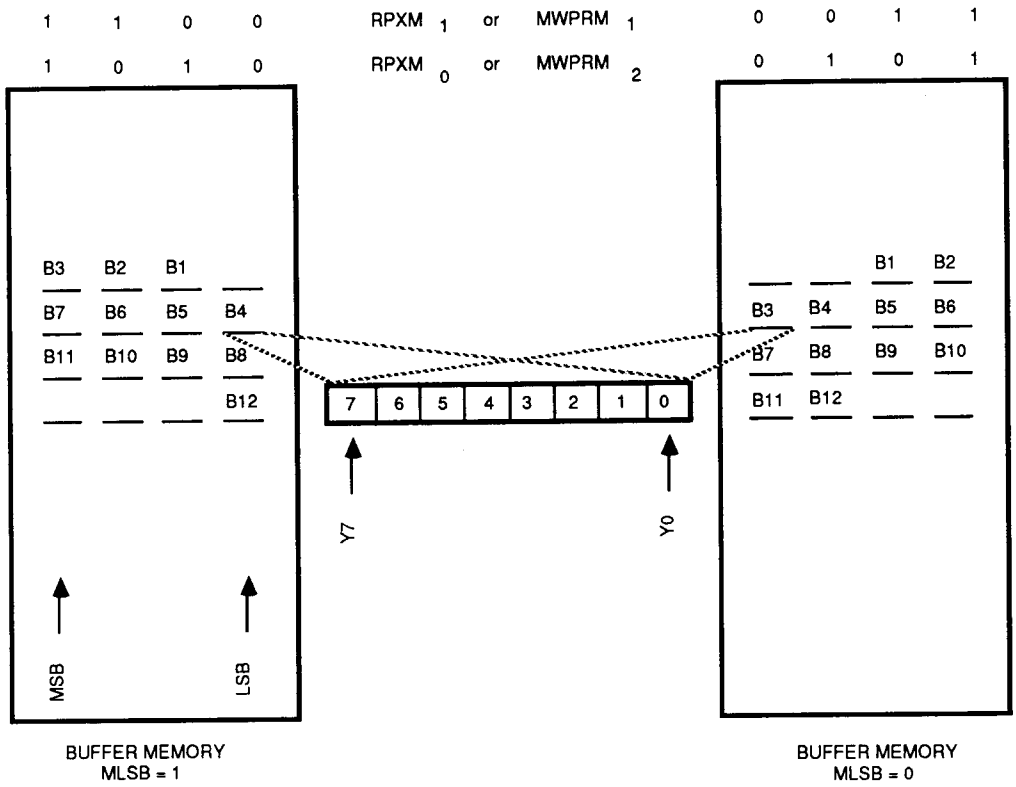
Parity is generated only when the DPC writes a received frame into memory. Parity is checked for all other transfers, including NP and Host read and write operations. Any parity error is flagged to the RBC using the PARERR pin.

$MWPRM_1$, $MWPRM_2$ (Mode for the WPR Byte Boundary Start). These bits indicate where to begin loading the received frame in Buffer Memory. This is shown in Figure 2. Since the frames can end on any byte of a long word, the frame may end on any boundary. The next frame is loaded at the byte boundary indicated by $MWPRM_1$ and $MWPRM_2$. These bits are useful for ensuring that the data unit for the next-higher layer in the protocol always begins on a long-word boundary so that byte realigning is avoided.



09730–005A

Figure 1. DPC Mode Register (MODE)



THE NUMBERS WITH A "B" PREFIX INDICATE THE ORDER IN WHICH BYTES ARE RECEIVED FROM OR TRANSMITTED TO THE MAC

09730-006A

Figure 2. Byte Ordering by MLSB

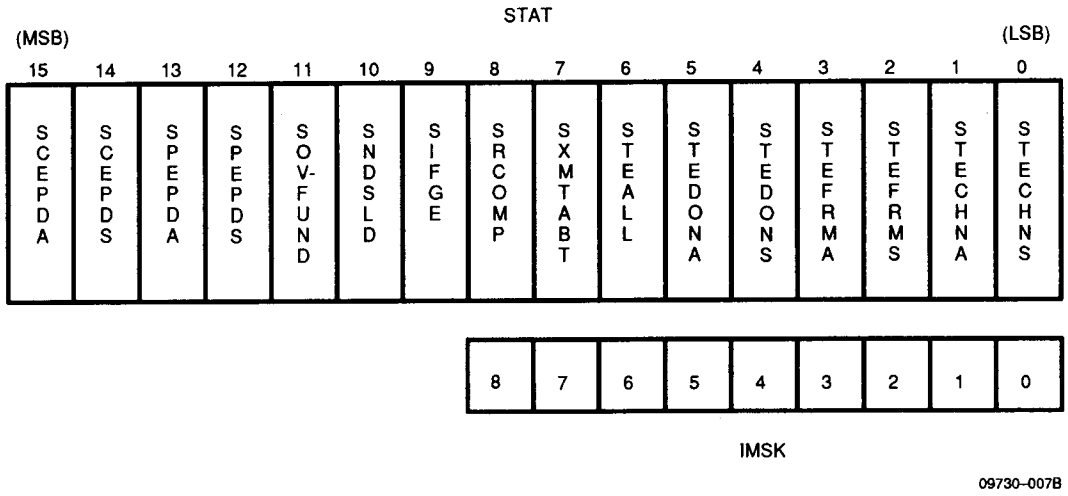


Figure 3. System Status Register (STAT) and Interrupt Mask Register (IMSK)

System Status Register (STAT) and Interrupt Mask Register (IMSK)

The NP status logic consists of a 16-bit register (STAT) that can be read by the NP on the NP-bus. These registers accumulate system-level status. The status bits accumulate in STAT until they are read through the IRDSTAT instruction.

Some of the status bits in STAT have a mask bit which is stored in the Interrupt Mask (IMSK) register. Unless a mask bit is set, the corresponding status condition generates a maskable interrupt $\overline{\text{MINTR}}$ for these bits in STAT. A DPC reset clears all status bits and sets all the mask bits in IMSK.

Maskable Status Bits

STECHNS (Bit 0) (Transmit End of a Chain Segment of S-frames). After all frames in a chain segment of an S-frame queue are transmitted out to the FORMAC, this bit is set. The NP can use this to add another chain segment to the queue. The queue supports at most two pending chain segments at any time in each FIFO.

STECHNA (Bit 1) (Transmit End of a Chain Segment of A-frames). This bit is similar to STECHNS and applies to the A-frame chain.

STEFRMS (Bit 2) (Transmit End of S-frames). This bit is set for each S-frame transmitted. If the frame happens to be the last frame in a chain segment, then the STECHNS bit will also be set.

STEFRMA (Bit 3) (Transmit End of A-frame). This bit is set for each A-frame transmitted.

STEDONS (Bit 4) (Transmit End using the "Done" bit in the S-frame Queue). A bit in the transmit descriptor called XDONE permits bringing MEDREQS LOW for a few clock cycles even if more frames are in the queue. In an FDDI ring this causes the FORMAC to issue the token, thinking that nothing is queued-up since MEDREQS goes LOW. This feature is useful for the FDDI protocol for limiting the maximum length of S-frame transmission in one token opportunity. The STEDONS bit is set whenever the frame having the XDONE bit set in its descriptor has been transmitted. The NP can use this indication for queuing more S-frames.

STEDONA (Bit 5) (Transmit End using the "Done" bit in the A-frame Queue). This bit operates similar to STEDONS but applies to the A-frame queue.

STEALL (Bit 6) (Transmit End for All Chains). This bit is set when the DPC has finished transmitting every frame in every chain segment for both S-frame and A-frame queues.

SXMTABT (Bit 7) (Transmit Abort). This bit is set on a transmit abort caused by either of the following: 1) assertion of the XMTABTI pin; or 2) XMEDAVS or XMEDAVA going LOW when RDYTBYT is HIGH.

SRCOMP (Bit 8) (Receive Complete). This bit is set at the completion of frame reception following the writing of the frame status and length information and the zero long word at the end of the frame. Frames that are flushed by the FSHRCVF pin do not set this bit, but those aborted by the RCVABT pin do.

Non-Maskable Status Bits

SIFGE (Bit 9) (Inter-Frame-Gap Error). This bit is set when two frames are received too close to one another. The DPC has an internal FIFO which receives frames as bytes on the Y-bus and unloads long words (32 bits) into the Buffer Memory. Status and length of the frame may also be written in front of the frame. If another frame is received before these operations are completed, a SIFGE is generated. The DPC requires an inter-frame gap of at least 6 bytes. In certain states, it can operate with less. The DPC only sets SIFGE and asserts the MISFRM pin when its FIFO cannot store the frame being received.

SNDSLD (Bit 10) (NP and DPC Simultaneous Load). This bit is set when the DPC and NP simultaneously attempt to load any register in the DPC.

SOVFUND (Bit 11) (Overflow and Underflow). This bit is set if, due to a malfunction external to the DPC, the DPC does not receive the acknowledges to its read and write requests, and its internal FIFO overflows or underflows. This condition should never occur if the RBC and DPC are functioning properly. If it does occur, the DPC must be reset to restart.

Note: If an internal FIFO underflow occurs during transmit, then the following happens:

1) Both transmit queues are cleared;
2) XMTABTO is asserted; and
3) SOVFUND is set.

SPEPDS (Bit 12) (Parity Error in Pointer or Descriptor of an S-frame Queue). This bit is set if the DPC detects a parity error while reading link-list control information, such as a pointer or a descriptor. This indicates that either the jump address to the next link or the length of the present frame may be corrupted. The transmit queue is flushed when this is detected. S-frame lock is set. This case also causes the S-frame transmit lock to be set, and therefore causes XMTABTO to be asserted.

SPEPDA (Bit 13) (Parity Error in Pointer or Descriptor of an A-frame Queue). This bit is similar to the SPEPDS, but applies to the A-frame queue. A-frame lock is set.

SCEPDS (Bit 14) (Coding Error in Pointer or Descriptor of an S-frame Queue). This bit is set if the DPC detects a violation of the required coding format for a pointer or a descriptor in the S-frame queue. This case also causes the S-frame transmit lock to be set, and therefore causes XMTABTO to be asserted.

SCEPDA (Bit 15) (Coding Error in Pointer or Descriptor of an A-frame Queue). This bit is similar to the SCEPDS, but applies to the A-frame queue. It causes the A-frame transmit lock to be set and XMTABTO to be asserted. The transmit queue is flushed when this bit is detected.

Buffer Memory Data Organization

The DPC manages the data transferred between the Buffer Memory and the FORMAC. Each transmit frame in the Buffer Memory consists of a descriptor at the first location containing length and control information about the frame. This is followed by the data being transmitted or received, and a pointer at the end to point to the next frame. Each received frame is stored in Buffer Memory with a status and length word immediately preceding the data received.

The first location of each frame in the Buffer Memory is a 32-bit descriptor field. Bits 0–15 contain the length of the frame. Bits 16–30 store the status of either a received frame or a frame to be transmitted. Bit 31 is used to flag the last frame in a chain segment for transmit, or the last frame received in a receive FIFO.

Receive Frame Descriptor (Status and Length)

The format of a descriptor for a received frame is shown in Figure 4. It is generated by the DPC and written to the Buffer Memory to become a part of the frame for use by higher-level protocols. The format of a descriptor for a frame to be transmitted is shown in Figure 6. The descriptor is not transmitted. Figure 5 shows how frame status and length are stored in Buffer Memory.

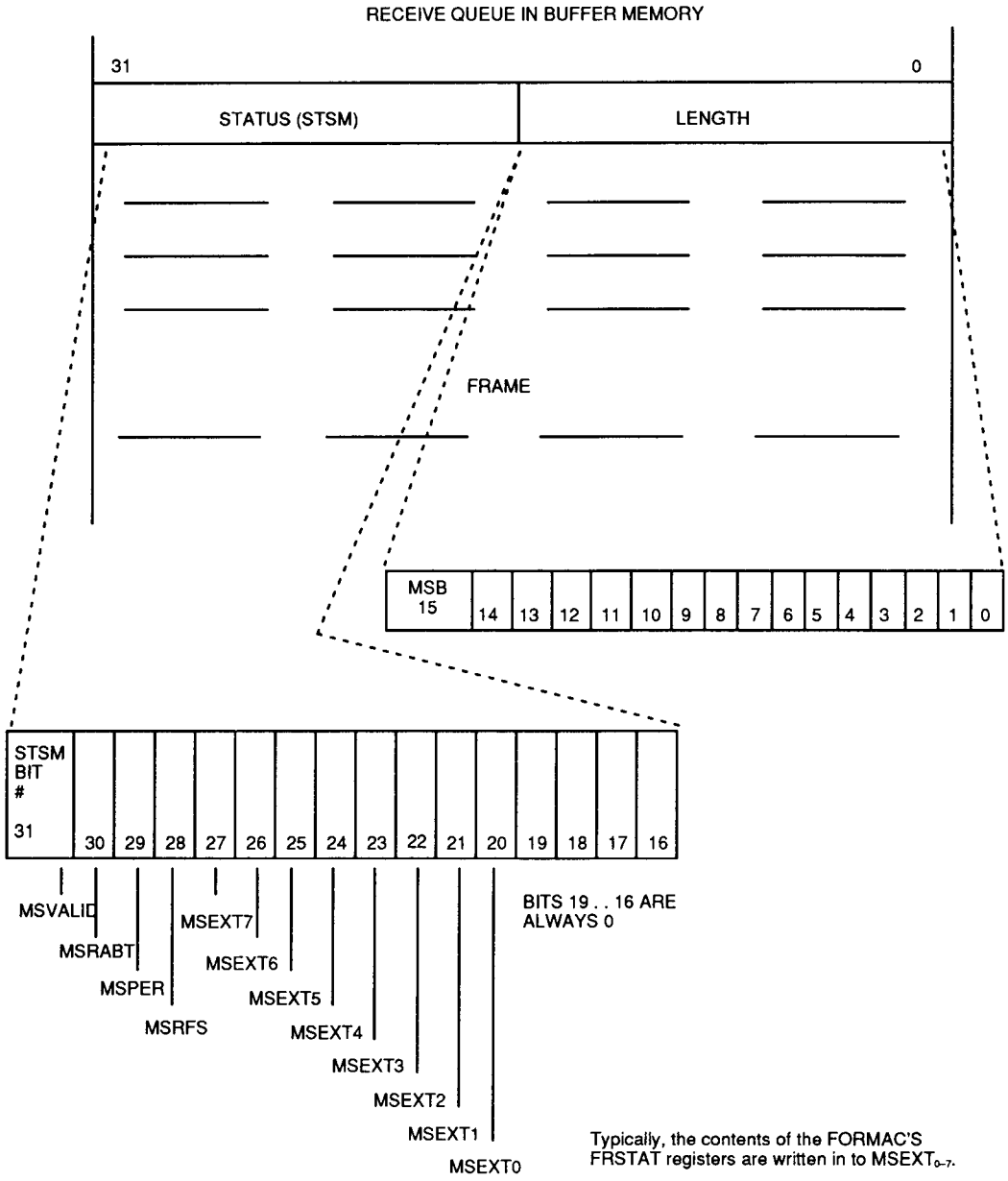
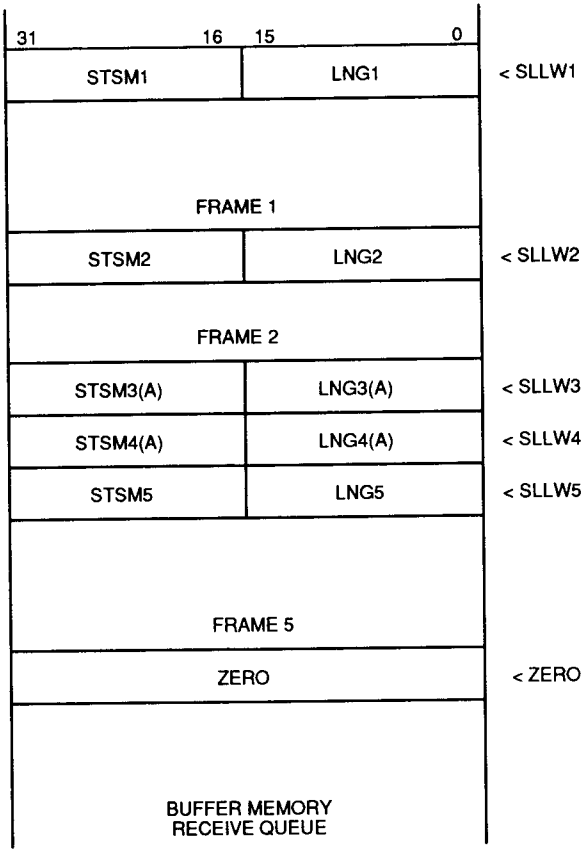


Figure 4. Format of a Receive Frame Descriptor



ZERO INDICATES BIT 31 = 0
FRAMES 3 & 4 WERE ABORTED BECAUSE OF VARIOUS ERROR CONDITIONS

09730-009A

Figure 5. Status and Length Storage in Buffer Memory

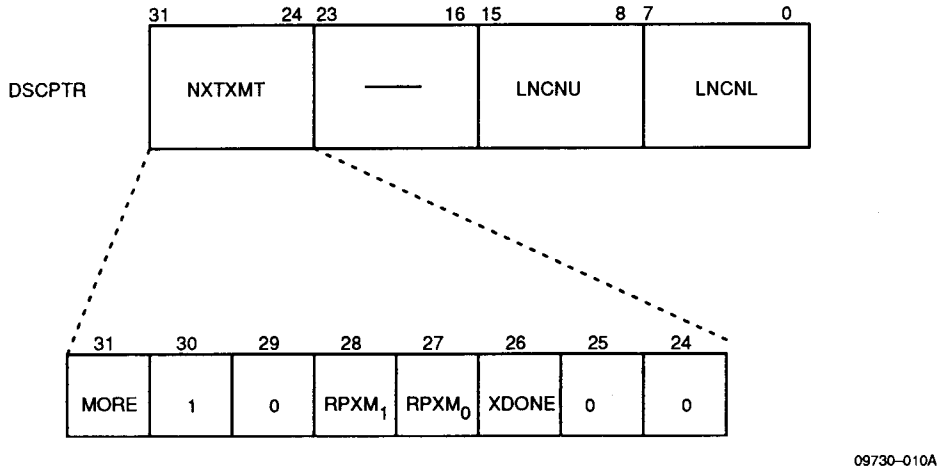


Figure 6. Format of a Transmit Frame Descriptor

Receive Frame Status (STSM)

Status is stored in STSM in buffer memory only after a frame is received and written into Buffer Memory. It is cleared from the DPC buffer after that frame and status has been written into memory.

Of this 16 bits of status, only 12 bits are used (bits 16–19 are always zero).

The STSM buffer in the DPC records status of the frame from the moment it enters the DPC until the status and length for this frame are written into the memory. Hardware outside the DPC (e.g., the FORMAC) could also generate status on the frame. If this status is provided on the Y-bus within two cycles of the end of the frame (RECEIVE going LOW), it will be included with the other frame status for storage in the Buffer Memory. If status is not provided to the DPC, then the status of the previous frame will be assigned to the current frame.

The individual bits of STSM are defined as follows:

MSVALID (Memory Status Valid). This bit is always a one. It is used for overwriting the "0" which indicates end of queue.

MSRABT (Memory Status for Receive Abort). This indicates that the frame was aborted. This could happen if the RCVABT or INICLBN pin goes HIGH during reception of data.

MSPER (Memory Status for Parity Error). This indicates that a parity error was detected when the frame was being received (the receive frame is still accepted since parity errors may be acceptable for some applications such as voice and FAX).

MSRFS (Memory Status for Receive Frame Short). This indicates that the received frame written in memory is short. It missed some bytes at the beginning because this frame was received too soon after the previous received frame (even though it is short, the receive frame is still accepted).

MSEXT$_{0-7}$ (Memory Status from External Source). These status bits are interpreted as defined by the user. Typically, the status generated in the FORMAC's Frame Status Register (FRSTAT) can be dumped on the Y-bus following the frame (within the next two BCLK cycles), and the DPC writes it into STSM as these bits. Examples of status conditions that the FORMAC can pass on this way include CRC errors, logical or physical address match, etc. Refer to the frame Status Register Section of the 79C83 data sheet.

Transmit Frame Descriptor

Figure 6 shows the fields within a descriptor for a transmit frame. The positions of the bits are independent of the MLSB bit in the mode register. The NXTXMT byte contains control information about handling the frame. The bits within NXTXMT are explained below.

MORE. This bit, if one, indicates that it is not the last frame in this chain segment.

RPXM$_{0-1}$. These two bits specify the byte boundary of the first byte of the frame. Figure 2 shows their interpretation, which also is a function of the MLSB bit.

XDONE. Normally the DPC keeps MEDREQS (or MEDREQA) HIGH when MORE = 1. The XDONE bit, when zero, keeps this operation unaffected. When

XDONE = 1, then even if MORE = 1, the DPC de-asserts MEDREQS (or MEDREQA) for three BCLK cycles. If MORE = 1 when this happens, MEDREQS (or MEDREQA) is asserted again after three cycles. This bit is used for controlling the number of S-frames sent per token in FDDI.

Frame Length

The LNCNU and LNCNL together form a 16-bit word that represents the length of the frame in bytes. This 16-bit word is loaded into the length counter (LNCN) register when the DSCPTR is read in the transmit mode. The length has to be non-zero. In the receive mode, after a frame has been written into memory, the length is written into the descriptor from the LNCN register.

The LNCN contains the number of bytes transmitted or received on the Y-bus. In FDDI applications where the DPC is used with a FORMAC, this count includes the FC, DA, SA, and INFO fields of the transmitted frame, but not the FCS, ED, and FS fields, since these are appended by the FORMAC. The count for received frames includes the FCS field, as well as the FC, DA, SA, and INFO. Thus, for a given size of frame, the LNCN for a received frame is 4 bytes greater than the LNCN for a transmitted frame.

Pointer Format

Figure 7 shows the fields within a pointer in a memory word. The positions of the bits are independent of the MLSB bit in the mode register.

The NXTXMT byte for a PNTR is a fixed pattern. This byte must be 10100000 (A0 in Hex) for upwards compatibility with future implementations.

The RPXU and RPXL fields in the pointer together form 16 bits. They contain the long-word address of the descriptor of the next frame. For S-frames, these are loaded into the RPXS in the RBC. For A-frames, they are loaded into the RPXA in the RBC.

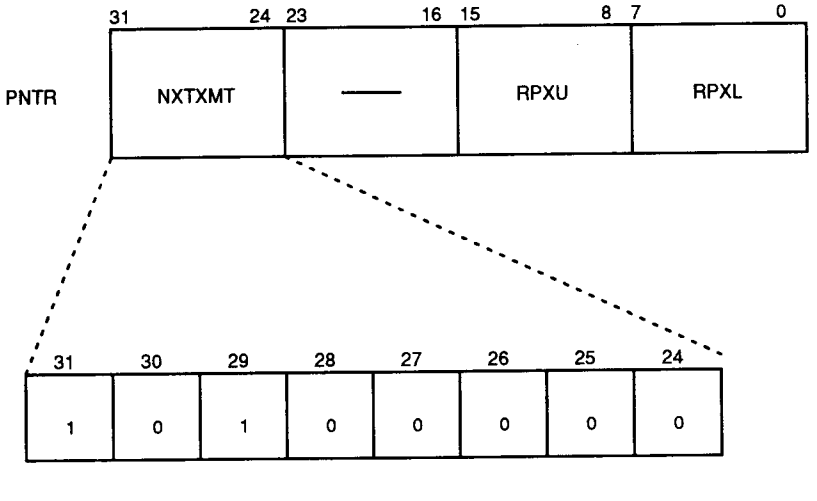


Figure 7. Format of a Frame Pointer

Byte Order Within a Word

The DPC provides two organizations for bytes within long words in memory. As shown in Figure 2, when the MLSB bit in the mode register is set, it indicates that the least-significant byte in the long word should be transmitted first. If the MLSB bit is reset, the most-significant byte is transmitted first. However, the order of bits in a byte is not changed. So for either value of the MLSB bit, 0 is provided on the lowest line and bit 7 on the highest; e.g., during transmit, bit 0 is provided on Y_0 and bit 7 on Y_7.

Byte Boundary Operation

The Buffer Memory is organized as 32-bit long words, and all the pointers in the RBC point to 32-bit long words. However, in many instances it becomes necessary to transmit a frame starting on a byte boundary which is not aligned with the 32-bit long word boundary. This happens in some cases when a header is added to a basic block of data to construct a frame for transmission. Two bits in a field called NXTXMT stored with every frame descriptor are used to determine byte boundary start. These are $RPXM_1$ and $RPXM_0$.

In a similar way, received frames can be stored starting at arbitrary byte boundaries using $MWPRM_1$ and $MWPRM_0$ bits in MODE which act as least-significant bit extensions to the WPR pointer of the RBC. The aligning of frames on a byte boundary is done within the DPC. Since the Buffer Memory can only be written in the long-word mode, a complete long word is written by the DPC, but during the first write transfer a few of the bytes may be garbage bytes. The start-byte boundary for received frames cannot be changed on a per-frame basis; it remains the same for all frames.

The NP and Host can only transfer data on long-word boundaries. Any stripping of garbage bytes would typically be done in software. It may be noted that even though there may be garbage bytes in any long word, all bytes must have valid parity to prevent generation of parity errors where parity is used.

RBC — DPC Interface

The DPC interacts with the RBC by transferring frames received from the media into the Buffer Memory receive FIFO, and by transferring frames to be transmitted from the Buffer Memory S-frame chain FIFO or A-frame chain FIFO to the media. During the transfer, the DPC converts the received data from 8-bit bytes to 32-bit words, and the data to be transmitted from 32-bit words to 8-bit bytes. At any time the DPC can either receive or transmit, but not both simultaneously. A full-duplex system can be built with two RBCs, two DPCs, and two Buffer Memories.

Received Data

The DPC transfers received data by issuing a DWRREQ request to the RBC for each long word to be transferred. When serviced, the RBC enables the Buffer Memory to write a long word of data present on the D-bus by using the address contained in the RBC pointer called WPR. After writing the location, the RBC increments WPR, setting up the pointer for the next word to be written. When WPR equals EAR, then it wraps around to the SAR value.

For received frames (DPC write requests), the DPC does the housekeeping for each frame. It accumulates status information and frame length, which it sends to the Buffer Memory at the end of each normal frame transfer (mode register bit MSLNSTM set HIGH). The RBC adds this data to the beginning of the frame in Buffer Memory in the following manner: the WPR, under RBC control, skips a memory location at the start of the frame storage area and returns to this location to store the status and length information after the frame is transferred.

The timing relationships of the various signal lines involved in receiving a frame (MSLNSTM bit is HIGH) is shown in Figure 9 of the Am79C81A data sheet.

Data To Be Transmitted

Data to be transmitted using FDDI standards is divided into two categories, S-frames and A-frames. Each category is stored in the Buffer Memory in a separate chain FIFO. Through the use of the THT timer and T– late latch, the FORMAC can determine whether to access the S-frame chain or the A-frame chain for the next transmission.

Individual frames may be assigned to any memory location in the Buffer Memory by the NP. However, every long word within the frame must be in contiguous memory locations. The first 32-bit word in the frame is a field called a descriptor (DSCPTR). The last 32-bit field immediately following the data is called a pointer (PNTR). The DSCPTR includes control information and the length of the frame in bytes. The PNTR contains the address of the next frame (a pointer to its DSCPTR).

One or more frames linked together is called a chain queue. Chains can be linked together by making the pointer of the last frame in one chain point to the beginning of another chain. The NP transfers frames into the Buffer Memory as chain segments. The last frame in a chain queue is flagged by setting the MORE bit of the descriptor equal to zero. Figure 2 of the Am79C81A data sheet shows how multiple chain queues are configured in memory.

The DPC transmits one S-frame for each XMEDAVS or one A-frame for each XMEDAVA until the entire chain queue is transmitted. It stops after the frame with the MORE = 0. The DPC assumes that no frame, DSCPTR, or PNTR associated with the frames in a chain segment will be modified after the IENXMTS or IENXMTA instruction is given. After an IENXMTS or IENXMTA is given, the NP may add more frames to the S-frame chain FIFO or A-frame chain FIFO by creating a second chain segment and issuing another IENXMTS or IENXMTA instruction. The DPC can pipeline and remember up to two IENXMTS and two IENXMTA instructions.

When the DPC encounters the MORE = 0 at the end of a chain segment, if another chain queue has been queued-up and enabled by the NP, the DPC treats this as a continuation and keeps MEDREQ HIGH as if MORE were one.

All Buffer-Memory access requests from the NP, DPC, and Host arriving at the RBC are serviced on a first-come, first-served basis. However, if the NP or Host and the DPC try to access the memory at the same time, then the DPC is serviced and the other request is placed in a pending queue. The DPC can issue a DISNHRQ to disable NP and Host requests when necessary.

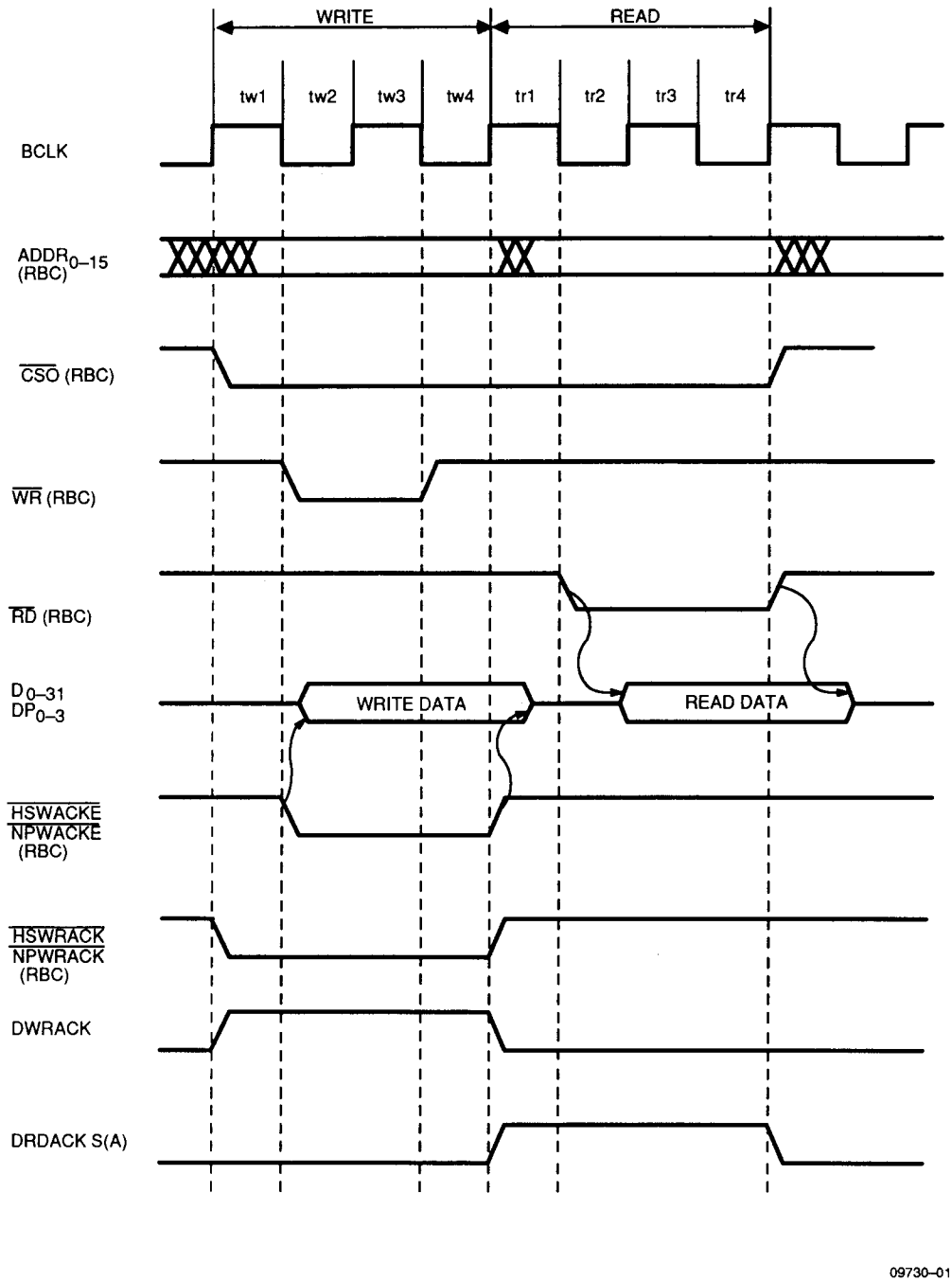


Figure 8. Buffer Memory — DPC Data Transfer Timing

S-frame Transmission

After the NP has transferred a chain segment into the S-frame FIFO, it issues an IENXMTS to DPC to enable the DPC to transfer one chain segment of S-frame data. The DPC issues a request (MEDREQS) to the FORMAC to transmit S-frames. When the FORMAC receives a token from the media, it acknowledges the DPC request to send S-frames by issuing XMEDAVS to the DPC. The DPC then issues a DRDREQS and an LDRPXS to make the RBC load the pointer to the first frame into the RPXS register. The RBC uses the pointer RPXS for the S-frames. For S-frame transmission, the DPC must issue a DRDREQS request for each long word in each frame. As with DWRREQ, the RBC takes one or two clocks to acknowledge DRDREQS. RPXS is incremented automatically by the RBC after each long word is transferred unless an LDRPXS was issued with the last request.

The DPC must know the length of a frame to determine when the end of a frame is reached. This information is in the descriptor at the start of the frame. When the end of a frame is reached, the DPC must issue an LDRPXS along with the DRDREQS. This lets the RBC know that the RPXS is pointing to the last location in the frame. This location contains the address of the start of the next frame. This pointer is read on the next XMEDAVS.

To get the RBC to load a new value into its RPXS, the DPC asserts and then de-asserts DRDREQS and LDRPXS simultaneously. After that cycle, RPXS gets its new value from the D-bus to start reading a new frame. The RPXS is incremented after each word is transferred until the RBC again receives a DRDREQS and LDRPXS simultaneously.

The timing diagram to transmit an FDDI S-frame in chain queue transmit mode (MENCHN = 1) is shown in Figure 13 of the Am79C81A data sheet.

A-frame Transmission

After the NP has transferred a chain segment into the A-frame FIFO, it issues an IENXMTA to the DPC to enable the DPC to transfer one chain segment of A-frame data. The DPC issues a request (MEDREQA) to the FORMAC to transmit A-frames. When the FORMAC receives a token from the media, it is ready to transmit. If the T–late flag is not set, it acknowledges the DPC request to send A-frames by issuing XMEDAVA to the DPC. The DPC then issues a DRDREQA to the RBC to transfer A-frames. The RBC uses the pointer RPXA for the A-frames. The DPC must issue a DRDREQA request for each long word in each frame. As with DWRREQ, the RBC takes one or two clocks to acknowledge DRDREQA. RPXA is incremented automatically by the RBC after each long word is transferred unless an LDRPXA was issued with the last request.

The DPC must also know when the end of a frame is reached so that it can issue an LDRPXA along with the DRDREQA. This lets the RBC know that the RPXA is pointing to the last location in the frame. This location contains the address of the start of the next frame. The RBC transfers this value to the RPXA pointer (RPXA is loaded from the D_{0-15} bus). If the LDRPXA is not issued with the DRDREQA, the RPXA is incremented after each read.

Claim/Beacon Mode

INICLBN and CLM/$\overline{\text{BEC}}$ both HIGH drive the RBC to the claim mode. See the "Claim/Beacon Mode" section of the Am79C81A data sheet for details.

FORMAC — DPC Interface

When the FORMAC receives data from the media, the DPC must be able to store this data immediately into the Buffer Memory to avoid losing some of the data. When the FORMAC receives a token, the DPC must furnish frames to the FORMAC from the Buffer Memory.

Receive Mode

Once initialization is complete and the receive lock is cleared (see below), the DPC is ready for the reception of frames. When the RECEIVE and DAVALID pins are asserted by the FORMAC, the DPC goes into Receive mode. In Receive mode, the DPC receives data on the Y-bus, stores it in its internal buffer FIFO, and generates DWRREQ to the RBC to write the received frame into Buffer Memory. During receive, no frame is transmitted, as the DPC is a half-duplex chip.

Receiving Normal Frames

Figure 9 shows the normal receive handshake between the DPC, FORMAC, and the RBC for a frame from the FORMAC.

Figure 10 shows the RBC–DPC handshake detail of a normal receive handshake. It also shows that two back-to-back write requests (DWRREQ) to the RBC are possible when the DPC internal FIFO has more than 4 bytes at the time that the FORMAC RECEIVE is de-activated.

Figure 11 shows that when RECEIVE is LOW but DAVALID is HIGH, the Y-bus data is interpreted as external FORMAC status and is stored in STSM as $MSEXT_{0-7}$ bits. The FORMAC provides this in the second cycle after RECEIVE goes LOW. A short interframe gap (IFG) is shown to illustrate that the DPC may ignore the short gap if it is ready for the next frame.

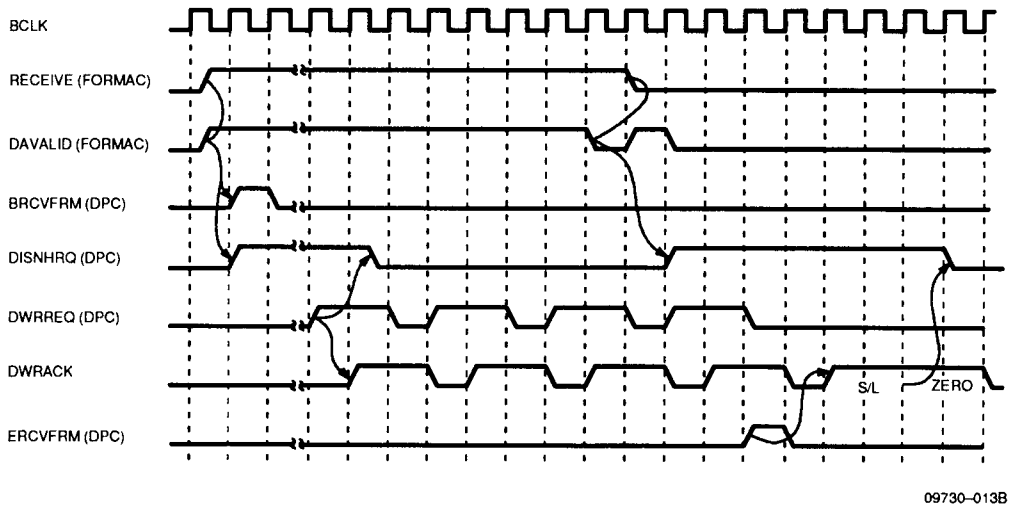
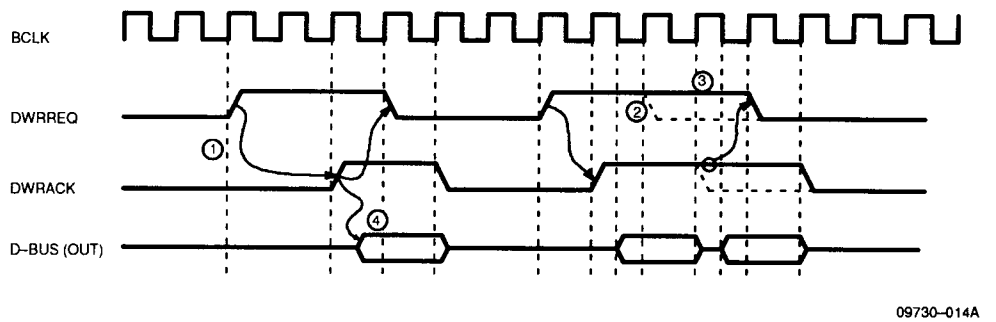


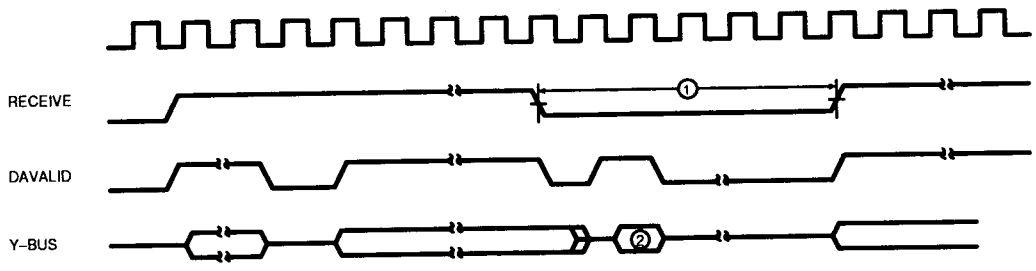
Figure 9. Normal Receive Handshake Between DPC, FORMAC, and RBC



Notes:

1. DWRACK should become active within two BCLK after DWRREQ active.
2. DWRREQ becomes inactive in one BCLK (in normal handshake) after DWRACK becomes active.
3. Two DWRREQs back to back are also possible (when RECEIVE is complete and the FIFO contains more than 4 bytes).
4. D–bus output data valid active.

Figure 10. RBC and DBC Handshake Detail of Normal Receive



09730–015B

Notes:

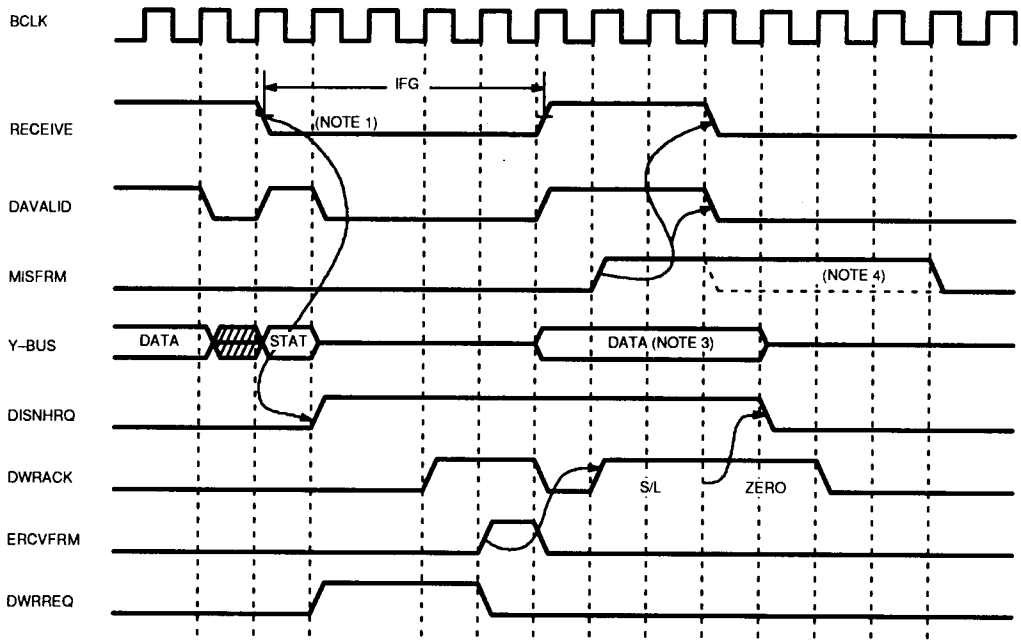
1. Inter–Frame Gap (IFG) less than eight (including "SD" and "FC") BCLKs violates the FDDI but may be ignored by the DPC.
2. External status (from FORMAC) to be stored into Descriptor in Buffer Memory.

Figure 11. Receive Normal Frame After Short IFG

Receiving Frame with Inter-Frame-Gap Error

The inter-frame gap (IFG) must be at least 6 bytes long to ensure storing the frames correctly in memory. Figure 12 shows a case where the IFG is less than 6 bytes. In some instances, the DPC cannot tolerate this and flags an error using MISFRM. If the DPC manages to successfully store the frame in spite of the IFG being too small, no error is flagged.

Errors due to short IFGs cause the SIFGE bit to be set in STAT, the MSRFS bit to be set in STSM in memory, and the MISFRM pin to be asserted to the FORMAC. Some initial bytes of the second frame are missed and the remainder are stored in memory.



09730-016B

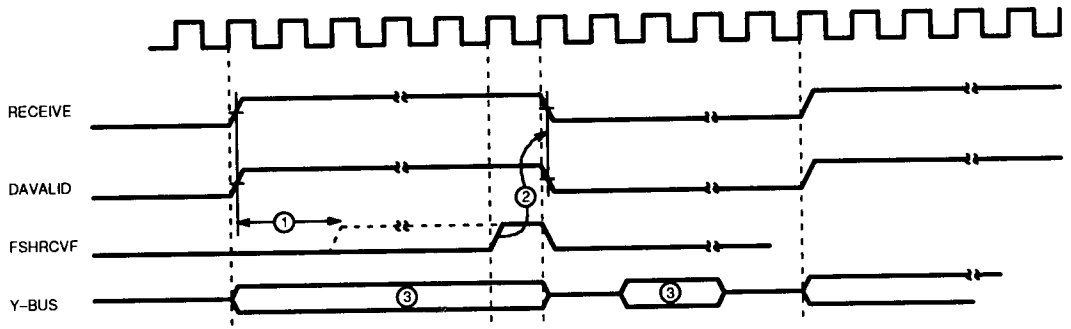
Notes:

1. Inter-Frame Gap of less than eight (including "SD" and "FC") BCLKs may cause IFG error.
2. When IFG error is detected, MISFRM is generated.
3. Data on Y-bus is ignored (not stored in memory) as long as MISFRM is active. If second frame as shown is less than eight BCLKs, then it could all be missed.
4. MISFRM is negated when DPC writes Zero from previous frame.

Figure 12. Receive Frame with Inter-Frame-Gap Error

Flushing a Receive Frame

Figure 13 shows the timing relationship between signals when flushing a frame. If FSHRCVF is asserted when RECEIVE is HIGH, the frame is flushed from the memory. This is effectively done by resetting the RBC pointer WPR. The FORMAC prevents FSHRCVF from going HIGH during the first cycle that RECEIVE is HIGH. FSHRCVF is HIGH for one clock cycle. The FORMAC also brings the RECEIVE and DAVALID signals LOW subsequent to FSHRCVF becoming HIGH. No status bits are set anywhere when FSHRCVF is asserted.



Notes:

1. FSHRCVF can become active one BCLK after RECEIVE and DAVALID become active. If it becomes active after RECEIVE becomes inactive, it is ignored by the DPC.
2. DAVALID and RECEIVE should go LOW after FSHRCVF becomes active.
3. DPC ignores this data.

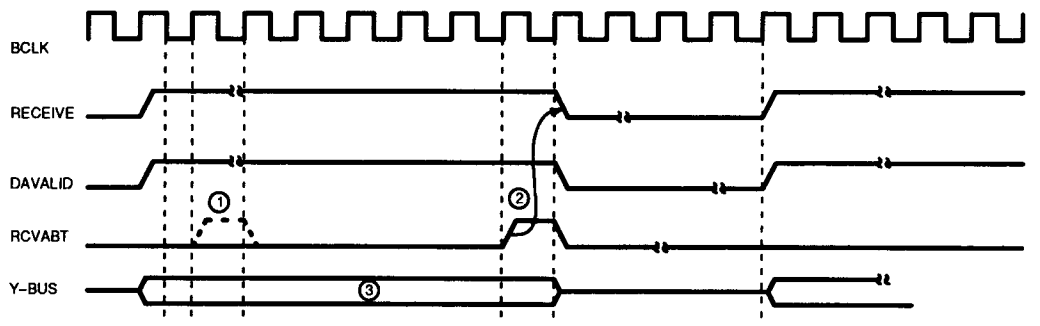
09730-017A

Figure 13. Flushing a Receive Frame

Aborting a Receive Frame

Figure 14 shows the timing relationship between signals when aborting a frame. This case is very similar to flushing a frame except that a status bit (MSRABT) is stored in STSM in Buffer Memory. The frame itself is not stored but the descriptor word (STSM and length) is written. The length stored for an aborted frame is 0.

If FSHRCVF and RCVABT occur simultaneously, then the flush function overrides the abort function. Neither signal should be asserted when RECEIVE is LOW. Multiple occurrences are allowed within the same RECEIVE window.



Notes:

1. RCVABT becomes active during RECEIVE active.
2. RECEIVE goes LOW after RCVABT becomes active.
3. Status of RCVABT (MSRABT) in status and length.

09730-018B

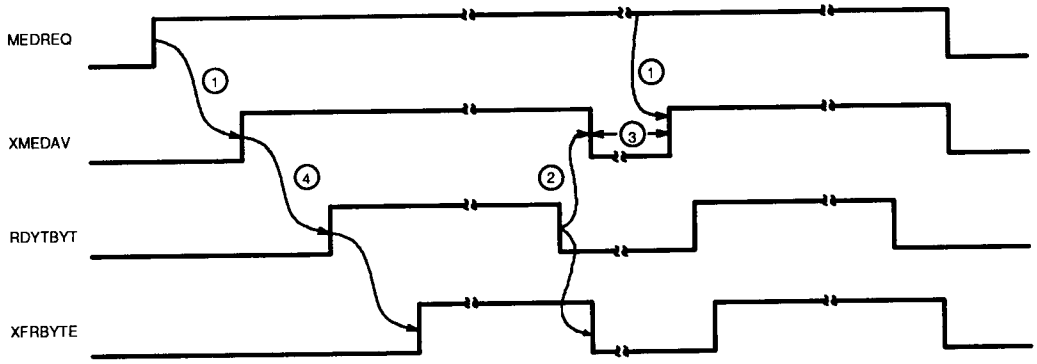
Figure 14. Aborting a Receive Frame

Transmit Mode

The DPC is in transmit mode only when a media request pin (MEDREQS and/or MEDREQA) and transmit media available pins (XMEDAVS or XMEDAVA) are active. A-frame transmission is independent with respect to the S-frame transmission. When both S-frames and A-frames are waiting, the FORMAC has to decide which to transmit. The FORMAC will not assert XMEDAVS and XMEDAVA simultaneously. Only one of the two can be asserted to indicate a transmit opportunity.

DPC — FORMAC Basic Transmit Handshake

The handshake for transmitting S-frames and A-frames is similar. Figure 15 shows such a handshake in its simplest form.



Notes:

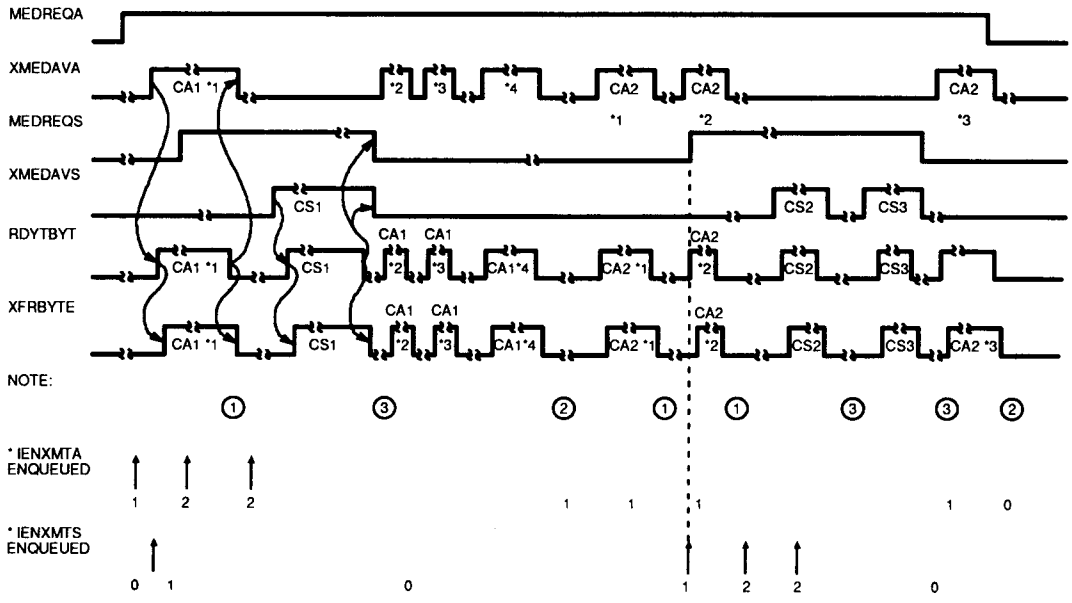
1. FORMAC receives token (decide on S-frames or A-frames).
2. End of one frame transmission.
3. XMEDAV stays LOW for at least one BCLK.
4. DPC has read one long word and has data to transfer to FORMAC.

09730-019B

Figure 15. DPC-FORMAC Basic Transit Handshake

Transmission of Multiple Chain Segments

Figure 16 shows how S-frame chain segments and A-frame chain segments can be multiplexed. The IENXMTSs (or IENXMTAs) are remembered up to a depth of two. Frame transmission is controlled by XMEDAVS (or XMEDAVA). Any chain segment may contain any number of frames.



1. End of one frame in A-frame Chain Segment
2. End of A-frame Chain Segment
3. End of one S-frame Chain Segment

CA: Chain of A-frames
CS: Chain of S-frames

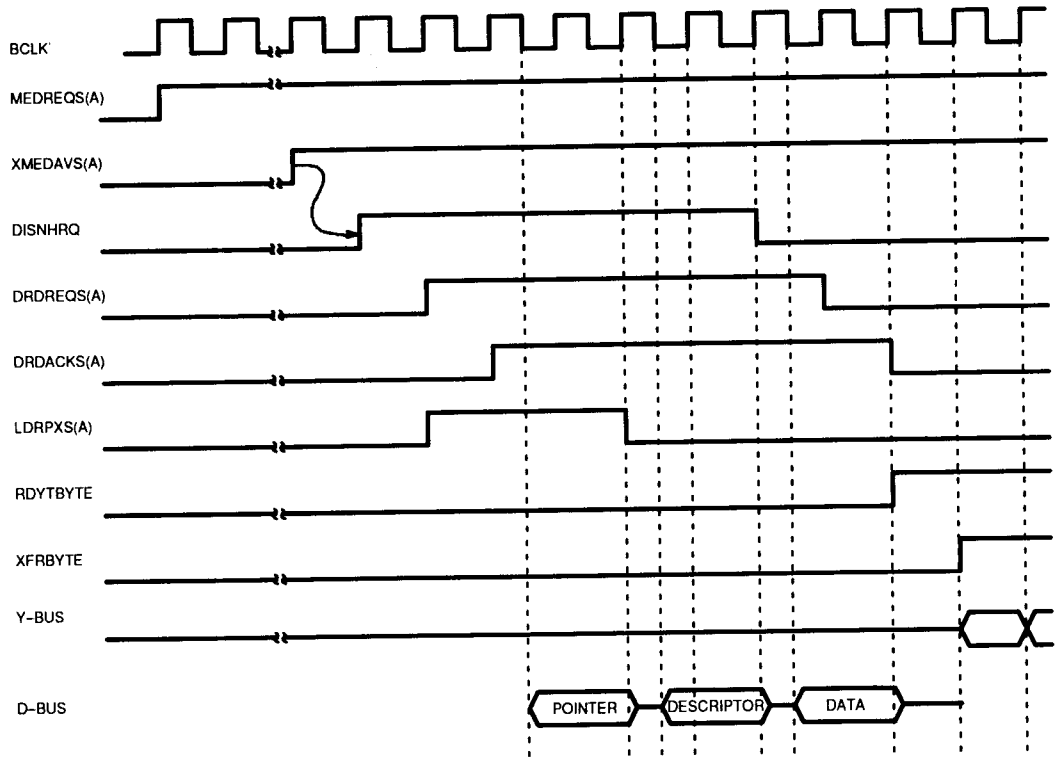
≀≀ = Possible Gap

09730-020B

Figure 16. Transmission of Multiple Chain Segments

Transmit Using the XDONE Control Bit

In the DSCPTR for every frame, the NXTXMT byte contains a bit called XDONE. This bit, if set HIGH, controls MEDREQS (or MEDREQA) and forces it LOW for three clock cycles at the end of a transmit. When this bit is LOW, the transmit operation is unaffected. The XDONE bit is used to limit the number of bytes transmitted in FDDI for S-frames and A-frames. This feature is provided so that the FORMAC can transmit the token immediately after a frame transmission or transmit (more) A-frames.



09730-021B

Figure 17. Transmit Sequence: Beginning of Frame

Additional Timing Sequences for Transmit

Figure 18 shows the timing sequence for the last frame to be transmitted. Figure 19 shows the timing sequence for transferring another frame for transmit. Figure 20 shows five sequences for a transmit abort.

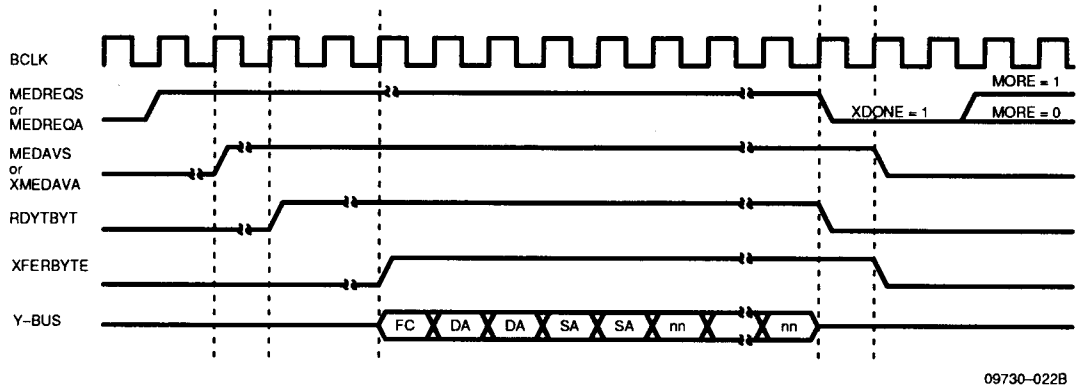


Figure 18. Transmit Last Frame in Queue

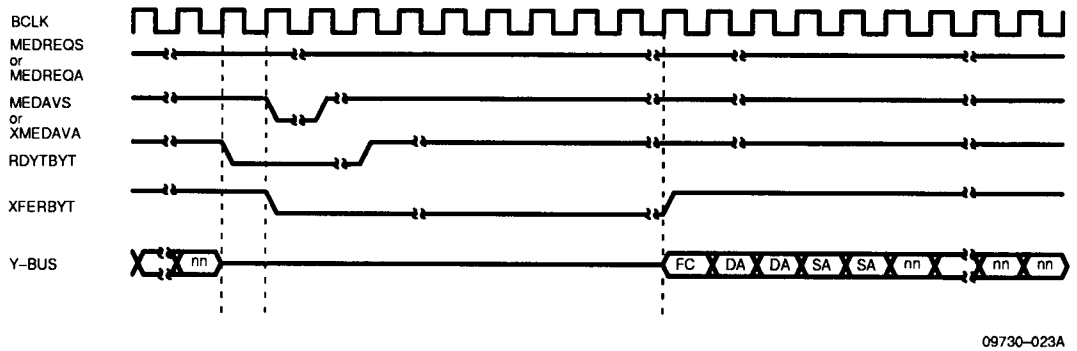
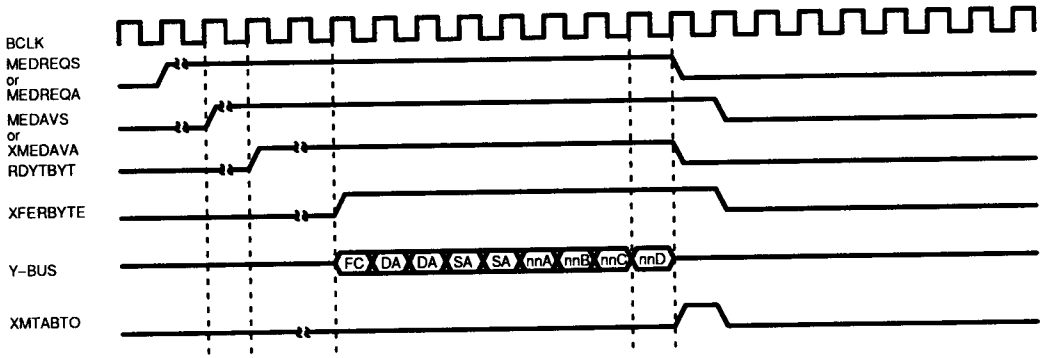
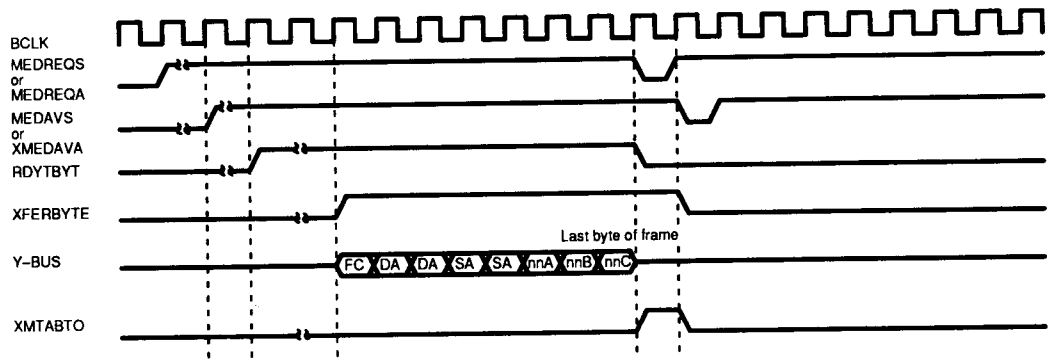


Figure 19. Transmit Another Frame



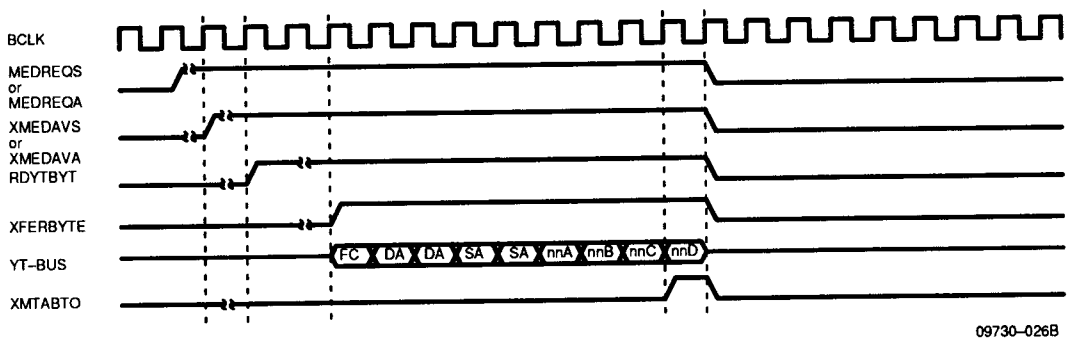
a) XMTABTO Due to NP Instruction to Am79C82A

09730-024B



b) XMTABTO Due to Parity Error During Data Read

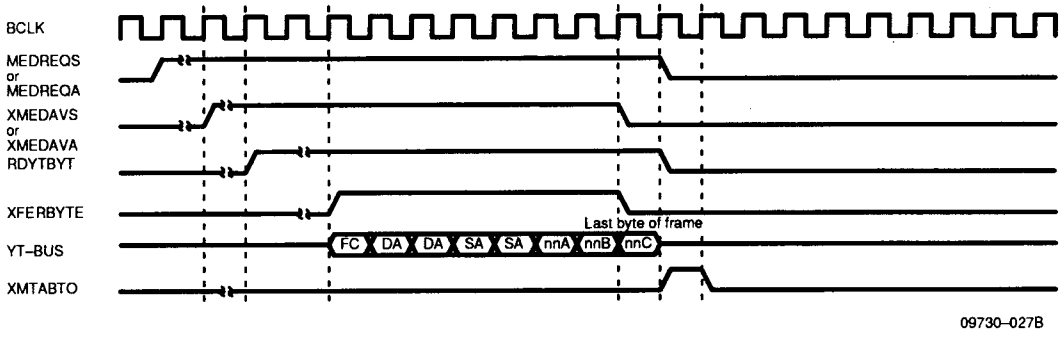
09730-025B



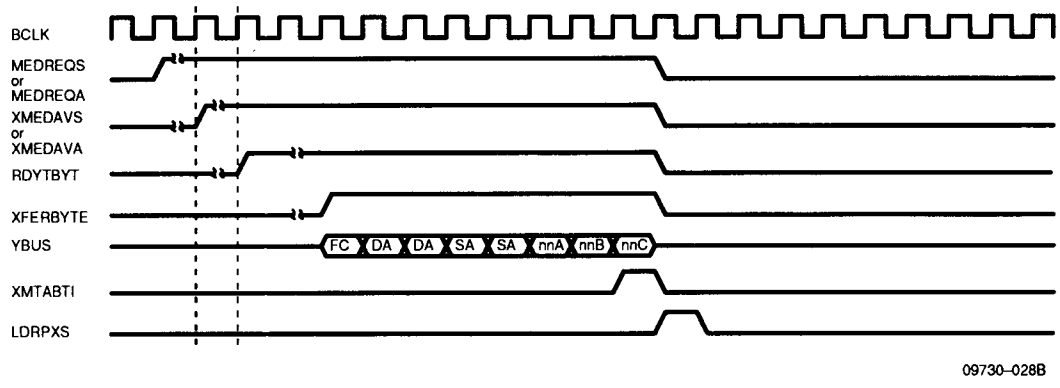
09730-026B

c) XMTABTO due to parity errors during pointer read, descriptor read, transmit FIFO underflow, or coding errors in pointers or descriptors

Figure 20. Transmit Abort Sequence



d) XMTABTO due to XMEDAVA(S) being driven LOW



e) XMTABTI From FORMAC

Figure 20. Transmit Abort Sequence, continued

ABSOLUTE MAXIMUM RATINGS

Storage Temperature	–65 to +150°C
Ambient Operating Temperature	–55 to +125°C
Maximum V_{CC}	–0.3 to +7.0 V
DC Voltage Applied to Any Pin	–0.5 to V_{CC} + 0.3 V

Stresses above those listed under ABSOLUTE MAXIMUM RATINGS may cause permanent device failure. Functionality at or above these limits is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

OPERATING RANGES

Commercial (C) Devices

Ambient Temperature (T_A)	0 to +70°C
Supply Voltage (V_{CC})	+4.75 to +5.25 V

Extended Commercial (E) Devices

Ambient Temperature (T_A)	–55 to +125°C
Supply Voltage (V_{CC})	+4.5 to +5.5 V

Operating ranges define those limits between which the functionality of the device is guaranteed.

DC CHARACTERISTICS over COMMERCIAL operating range

Parameter Symbol	Parameter Description	Test Conditions	Min.	Max.	Unit
V_{IL}	InputLOW Voltage			0.8	V
V_{IH}	Input HIGH Voltage		2.0		V
V_{OL}	Output LOW Voltage	I_{OL} = 4.0 mA		0.4	V
V_{OH}	Output HIGH Voltage (Note 1)	I_{OH} = –4.0 mA	2.4		V
I_{IX}	Input Leakage Current (Note 2)	0 V < V_{IN} < V_{CC}	–10	10	µA
I_{OZ}	Output Leakage Current (Note 3)	0.4 V < V_{OUT} < V_{CC}	–10	10	µA
I_{CC}	Power Supply Current	V_{CC} = Max. f(BCLK) = 12.5 MHz		160	mA

Notes:
1) V_{OH} does not apply to open-drain output pins.
2) I_{IX} applies to all input-only pins.
3) I_{OZ} applies to all three-state Output pins and bidirectional pins.

CAPACITANCE*

Parameter Symbol	Parameter Descriptions	Typ.	Unit
C_{IN}	Input Pins	15	pF
C_{IO}	Bidirectional Pins	15	pF

*Pin capacitance is characterized at a frequency of 1 MHz, but is not 100% tested.

SWITCHING CHARACTERISTICS over COMMERCIAL operating range (Notes 1 & 3)

Parameter Number	Parameter Description	Parameter Symbol	Min. (Note 2)	Max. (Note 2)	Unit
1	Clock Period	BCLK	80		ns
2	HIGH Pulse Width	BCLK	35		ns
3	LOW Pulse Width	BCLK	35		ns
4	Setup Time Before BCLK ↑	$\overline{CSI}$, $\overline{DS}$, $INST_{0-3}$, R/$\overline{W}$	45		ns
5	Hold Time After BCLK ↑	$\overline{CSI}$, R/$\overline{W}$	2		ns
6	Hold Time After BCLK ↑	$\overline{DS}$	6		
7	Hold Time After BCLK ↑	$INST_{0-3}$	10		ns
8	Hold Time After BCLK ↑	NP_{0-15}	15		ns
9	Setup Time Before BCLK ↑	NP_{0-15}	12		ns
10	R/$\overline{W}$ ↑, $\overline{CSI}$ ↓, $\overline{DS}$↓ (Whichever Occurs Last) Until NP-bus is Enabled (Synchronous Mode)	NP_{0-15}	3		ns
11	Signal Valid After BCLK ↑	NP_{0-15}		51	ns
12	Signal Invalid After BCLK ↑	NP_{0-15}	3		ns
13	Signal Invalid After $\overline{CSI}$ ↑ or R/$\overline{W}$ ↓ (Whichever Occurs First at the End of a Synchronous Read Cycle)	NP_{0-15}	3		ns
14	R/$\overline{W}$ ↓ or $\overline{CSI}$ ↑ (Whichever Occurs First at the End of a Synchronous Read Cycle) to Bus Inactive	NP_{0-15}		60	ns
15	Signal Valid After $\overline{DS}$ ↑ or $\overline{CSI}$ ↑ (Whichever Occurs First in Asynchronous Write)	NP_{0-15}	0		ns
16	Signal LOW After NP-bus Valid	$\overline{READY}$	(T1 – 20)		ns
17–19	Unused				
20	Setup Time Before $\overline{CSI}$ ↓ or $\overline{DS}$ ↓ (Whichever Occurs Last in Asynchronous Read/Write)	R/$\overline{W}$ ↓ (Write) or R/$\overline{W}$ ↑ (Read), $INST_{0-3}$	0		ns
21	Unused				
22	Pulse Width HIGH (Asynchronous Read or Write)	$\overline{DS}$, $\overline{CSI}$	1.5 x T1		ns
23	Hold Time After $\overline{DS}$ ↑ or $\overline{CSI}$ ↑ (Whichever Occurs First at the End of Asynchronous Read/Write)	R/$\overline{W}$ ↑ (Write) or R/$\overline{W}$ ↓ (Read), $INST_{0-3}$	0		ns
24	Signal LOW After $\overline{DS}$ ↓ or $\overline{CSI}$ ↓	$\overline{READY}$	(3 x T1)	(4 x T1) + 70	ns
25	Signal Disabled After $\overline{DS}$ ↑ or $\overline{CSI}$ ↑	$\overline{READY}$		40	ns
26	Bus Active After $\overline{DS}$ ↓ or $\overline{CSI}$ ↓ (Whichever Occurs Last in Asynchronous Read)	NP_{0-15}	0		ns
27	Bus Valid After $\overline{DS}$ ↓ or $\overline{CSI}$ ↓ (Whichever Occurs Last in Asynchronous Read)	NP_{0-15}		(3 x T1) + 48	ns

SWITCHING CHARACTERISTICS, (Continued)

Parameter Number	Parameter Description	Parameter Symbol	Min. (Note 2)	Max. (Note 2)	Unit
28	Signal Invalid After $\overline{DS}$ ↑ or $\overline{CSI}$ ↑ (Whichever Occurs First in Asynchronous Read)	NP_{0-15}	0		ns
29	Bus Disabled After $\overline{DS}$ ↑ or $\overline{CSI}$ ↑ (Whichever Occurs First in Asynchronous Read)	NP_{0-15}		35	ns
30	Setup Time Before $\overline{DS}$ ↓ or $\overline{CSI}$ ↓ (Whichever Occurs First in Asynchronous Write)	NP_{0-15}	–(T1 x 2 –20)		ns
31–32	Unused				
33	ACKONE to Data Valid (Note 4)	D_{0-31}, DP_{0-3}		48	ns
34	DWRACK, MDWRACK to Data Valid (Note 4)	D_{0-31}, DP_{0-3}		48	ns
35	Signal Valid After BCLK ↓	D_{0-31}, DP_{0-3}		48	ns
36	Signal Invalid After BCLK ↑	D_{0-31}, DP_{0-3}	7		ns
37	Output Enabled After BCLK ↓	D_{0-31}, DP_{0-3}	5		ns
38	Output Disabled After BCLK ↑	D_{0-31}, DP_{0-3}		55	ns
39	Setup Time Before BCLK ↑	DWRACK, DRDACKS, DRDACKA, MDRDACK, MDWRACK, RECEIVE, DAVALID, XMEDAVS, XMEDAVA	23		ns
40	Hold Time After BCLK ↑	DWRACK, DRDACKS, DRDACKA, MDRDACK, MDWRACK, RECEIVE, DAVALID, XMEDAVS, XMEDAVA	8		ns
41	Setup Time Before BCLK ↑	ACKONE	30		ns
42	Hold Time After BCLK ↑	ACKONE	8		ns
43	Signal Valid After BCLK ↑	PARERR		55	ns
44	Signal Invalid After BCLK ↑	PARERR	7		ns
45–46	Unused				
47	Hold Time After BCLK ↑	D_{0-31}, DP_{0-3}	10		ns
48	Setup Time Before BCLK ↑	D_{0-31}, DP_{0-3}	10		ns
49	Unused				
50	Output Valid After BCLK ↑	Y_{0-7}, YP		70	ns
51	Output Invalid After BCLK ↑	Y_{0-7}, YP	15		ns
52	Output Enabled After BCLK ↑	Y_{0-7}, YP	Note 1i		
53	Output Disabled After BCLK ↑	Y_{0-7}, YP		60	ns
54	Output Valid After BCLK ↑	DWRREQ, DRDREQS, DRDREQA, BRCVFRM, LDRPXS, LDRPXA, DISNHRQ, XMTABTO, RDYTBYT, MEDREQS, MEDREQA		55	ns
55a	Output Invalid After BCLK ↑	DWRREQ, DRDREQS, DRDREQA, BRCVFRM, LDRPXS, LDRPXA, DISNHRQ, XMTABTO	8		ns
55b	Output Invalid After BCLK ↑	RDYTBYT, MEDREQS, MEDREQA	11		ns
56	Setup Time Before BCLK ↑	XFRBYTE	40		ns
57	Hold Time After BCLK ↑	XFRBYTE	3		ns
58	Bus Enabled After XFRBYTE ↑	Y_{0-7}, YP	3	30	ns

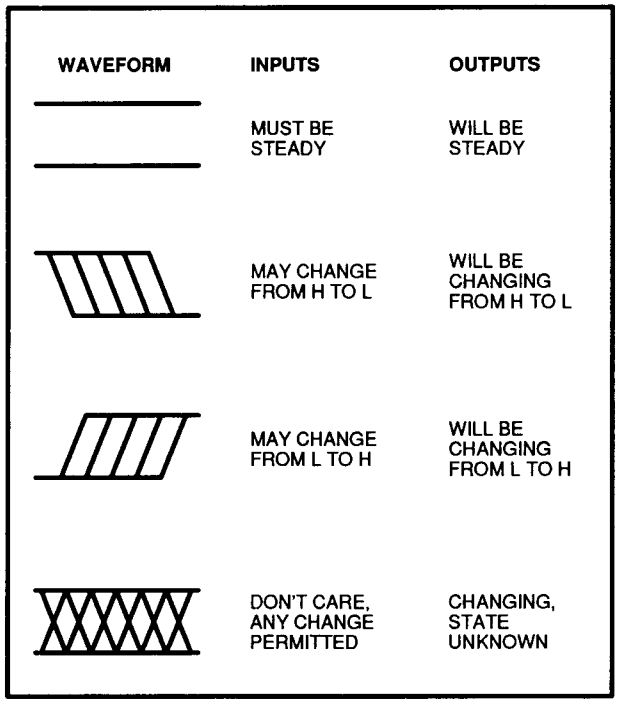
SWITCHING CHARACTERISTICS, (Continued)

Parameter Number	Parameter Description	Parameter Symbol	Min. (Note 2)	Max. (Note 2)	Unit
59	Bus Disabled After XFRBYTE ↓	Y_{0-7}, YP	2	45	ns
60	Setup Time Before BCLK ↑	INICLBN	20		ns
61	Hold Time After BCLK ↑	INICLBN	8		ns
62	Setup Time Before BCLK ↑	RBFERR, XMTABTI	25		ns
63	Hold Time After BCLK ↑	RBFERR, XMTABTI	10		ns
64	Output LOW After BCLK ↑	$\overline{\text{NMINTR}}$, $\overline{\text{MINTR}}$		45	
65	Output Disabled After BCLK ↑ (Note 1h)	$\overline{\text{NMINTR}}$, $\overline{\text{MINTR}}$	0		ns
66–69	Unused				
70	Setup Time Before BCLK ↑	Y_{0-7}, YP	10		ns
71	Hold Time After BCLK ↑	Y_{0-7}, YP	9		ns
72	Setup Time Before BCLK ↑	FSHRCVF, RCVABT	25		ns
73	Hold Time After BCLK ↑	FSHRCVF, RCVABT	6		ns
74	Signal Valid After BCLK ↑	ERCVFRM, MISFRM		55	ns
75	Signal Invalid After BCLK ↑	ERCVFRM, MISFRM	9		ns
76–79	Unused				

Notes:

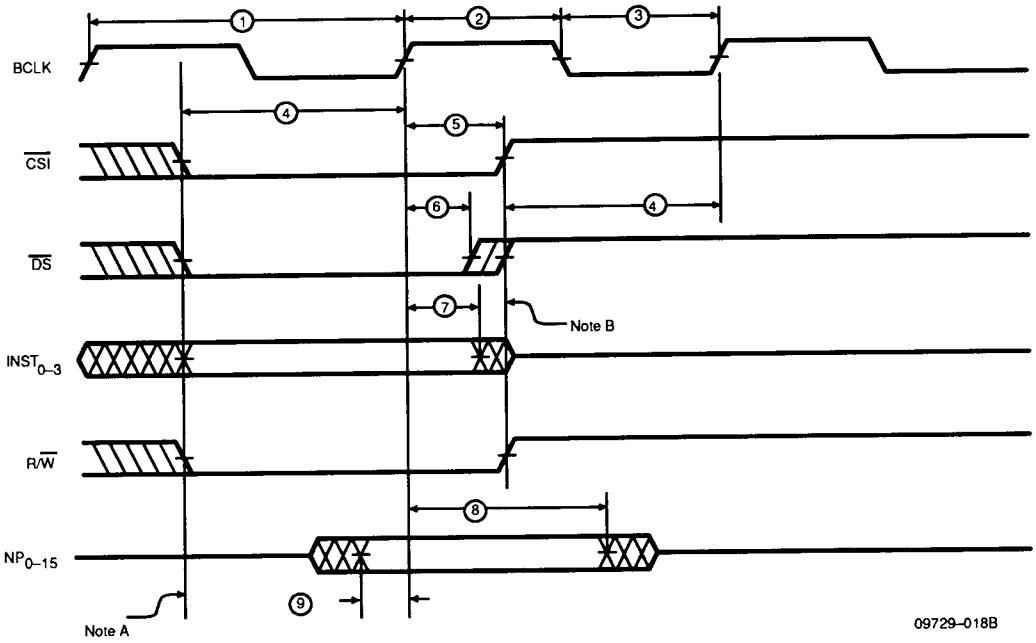
1. Measurement points for timing parameters are the following:
   a) Input waveforms: +1.4 V
   b) Output HIGH threshold: > 2.0 V
   c) Output LOW threshold: < 0.8 V
   d) Output valid: < 0.8 V or > 2.0 V
   e) Output invalid minimum: output is verified to be still valid at the minimum spec time.
   f) Output enabled: time when the output driver turns on.
   g) Output disabled: time when the output driver turns off.
   h) T65 is guaranteed by design.
   i) Guaranteed by T64 of FORMAC and T58 of DPC.
2. Numbered parameters such as T1 or T51 refer the switching characteristic number listed in the far left–hand column.
3. Maximum "BCLK to Signal Valid" delay and minimum "BCLK to Signal Invalid" delay specifications apply to both rising and falling edges of the signal listed, even though the measurement is only shown once in the Switching Waveforms section that follows. Valid and Invalid do not mean HIGH or LOW; see notes 1d and 1e and Switching Waveforms for the definition of Valid and Invalid.
4. Data Valid (D_{0-31}, DP_{0-3}) is controlled not only by the BLCK signal, but also the DWRACK/MDWRACK and ACKONE signals, whichever comes last.

Key to Switching Waveforms



09730-040A

SWITCHING WAVEFORMS

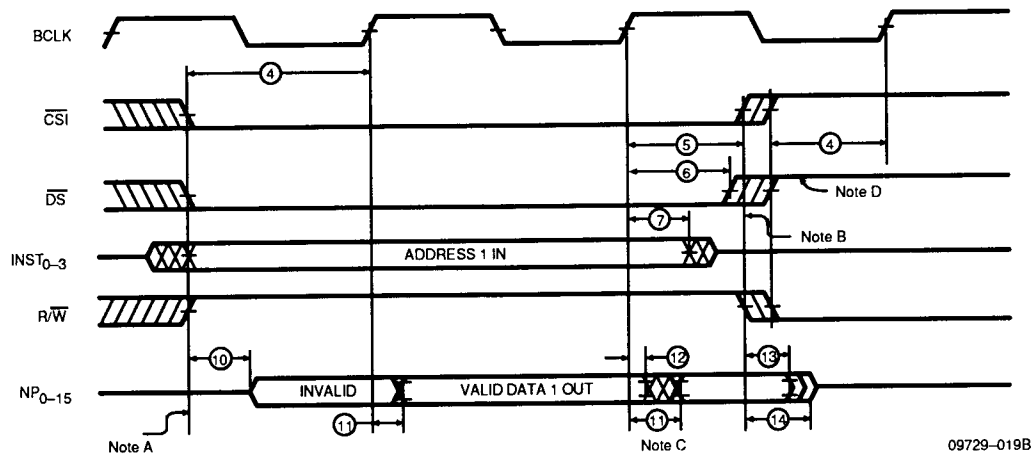


Notes:

A. Timing measured from falling edge of CSI, DS, or R/W or the assertion of INST $_{0-3}$, whichever occurs last.

B. Parameter #5 is measured from the rising edge of CSI, or R/W, whichever occurs first; Parameter #4 is measured from the falling edge of CSI, DS, or R/W, or the assertion of INST$_{0-3}$, whichever occurs last.

NP–bus Synchronous Write Timing



Notes:

A. Timing measured from falling edge of CSI or DS, or the rising edge of R/W (or the assertion of INST$_{0-3}$ for parameter #4 only), whichever occurs last.

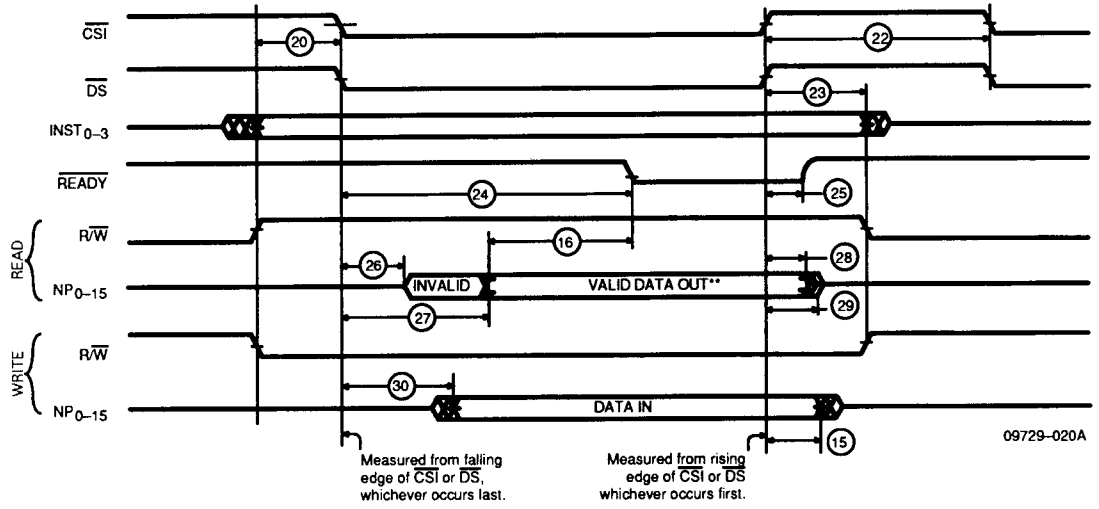
B. Timing is measured from the rising edge of CSI or the falling edge of R/W, whichever occurs first.

C. If an address register in the FORMAC is being addressed, then the register being read onto the NP–bus will be incremented with each rising edge of BCLK.

D. DS must be driven HIGH at the completion of each read cycle.

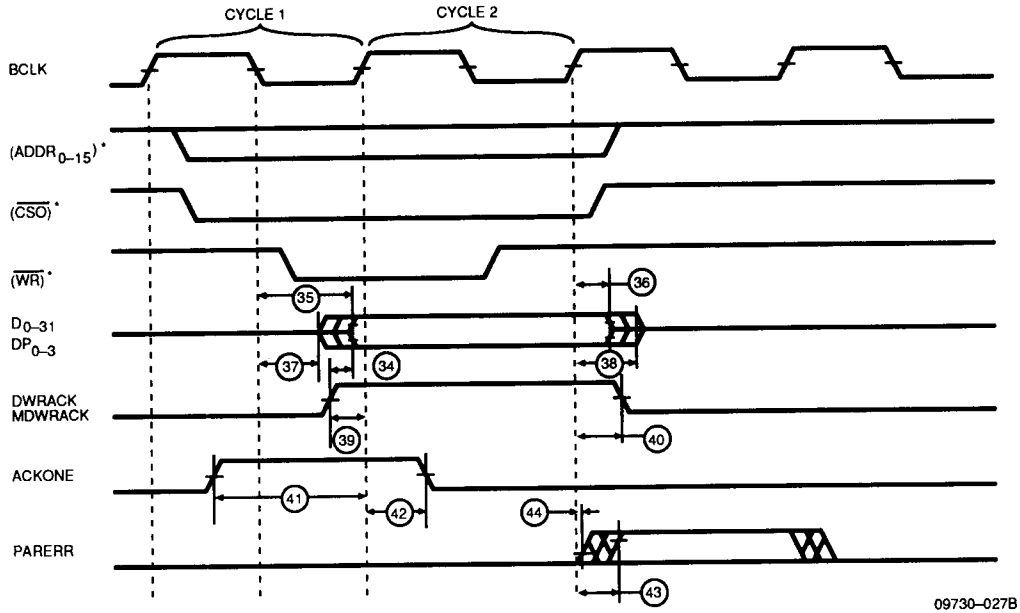
NP–bus Synchronous Read Timing

SWITCHING WAVEFORMS, (Continued)



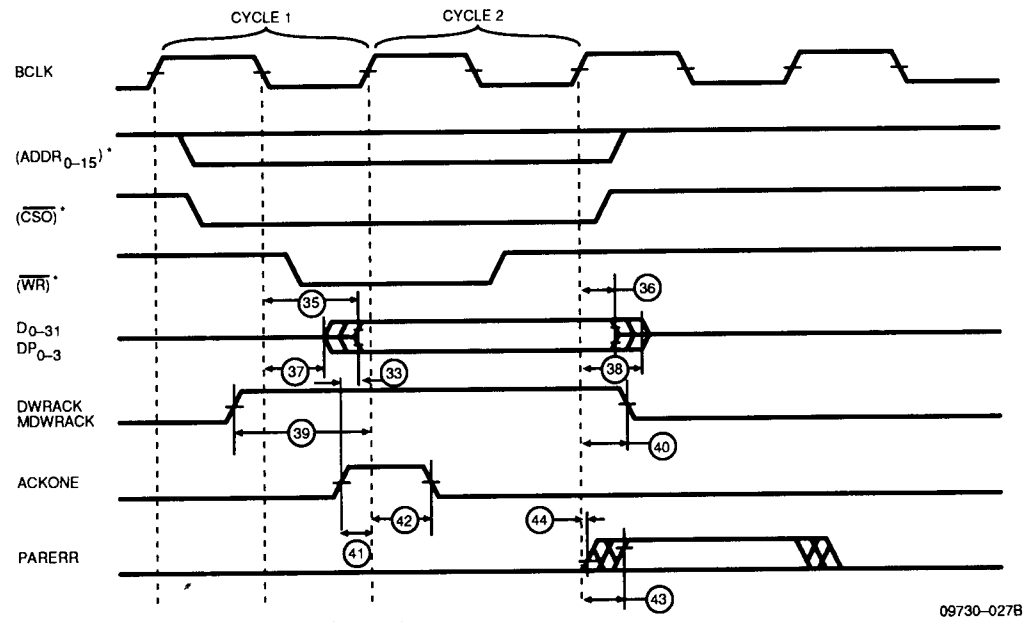
NP–bus Asynchronous Read/Write Timing

SWITCHING WAVEFORMS, (Continued)



*These Am79C81 RBC signals are shown for reference only.

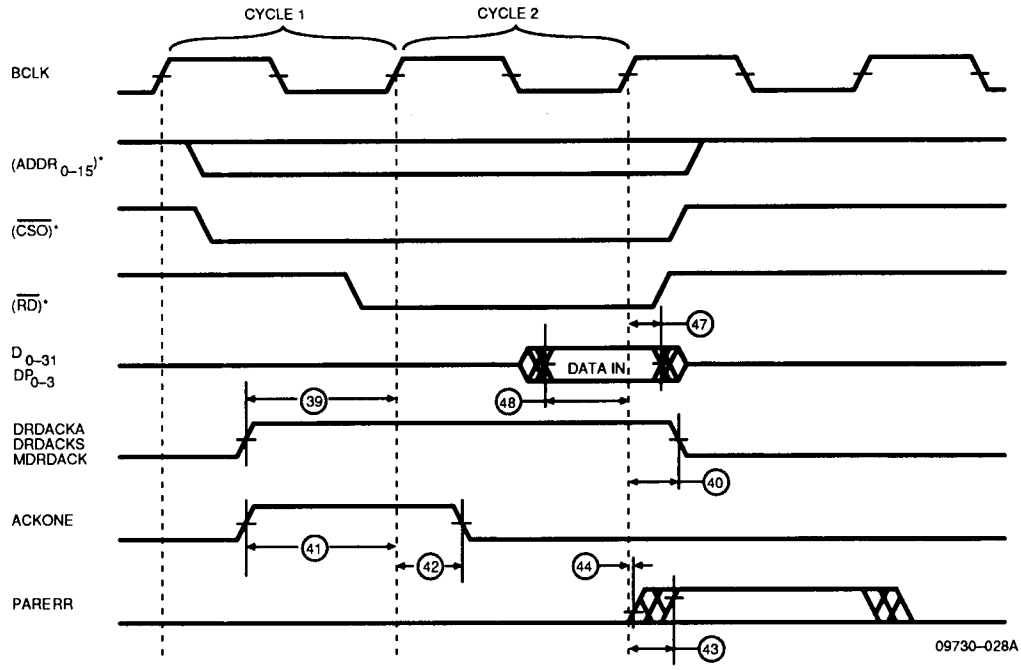
DPC Buffer Memory Write Timing I



*These Am79C81 RBC signals are shown for reference only.

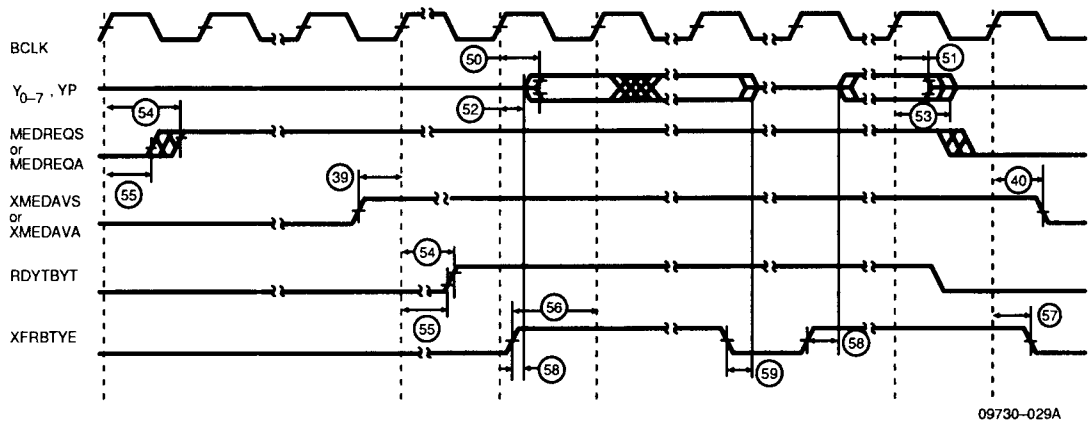
DPC Buffer Memory Write Timing II

Tom, Stacie will modify this drawing to match attached MacDraw version.



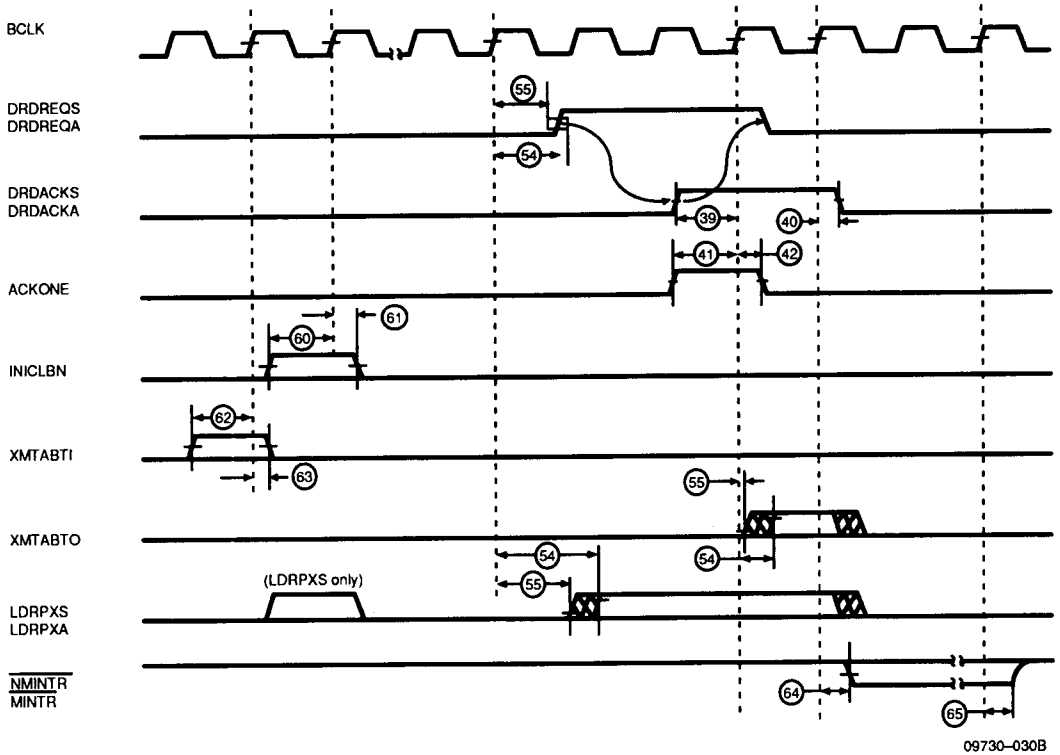
*These Am79C81 RCB signals are shown for reference only.

DPC Buffer Memory Read Timing

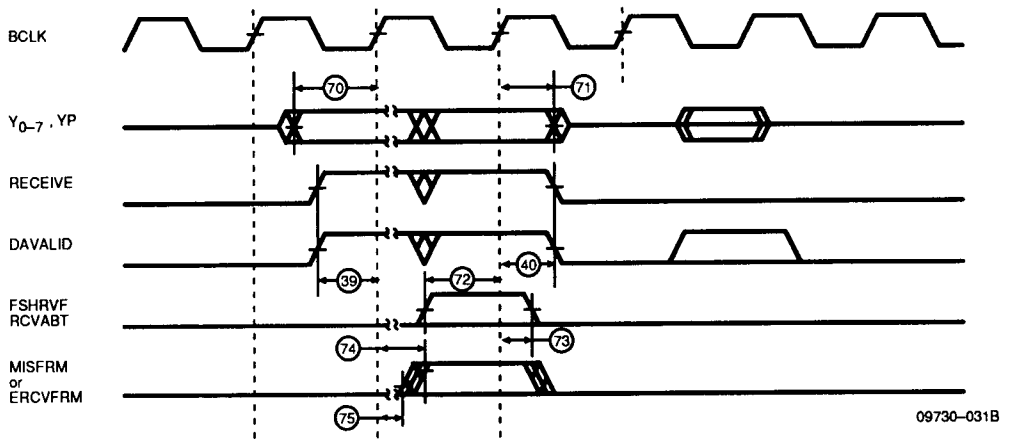


DPC Frame Transmit Timing I

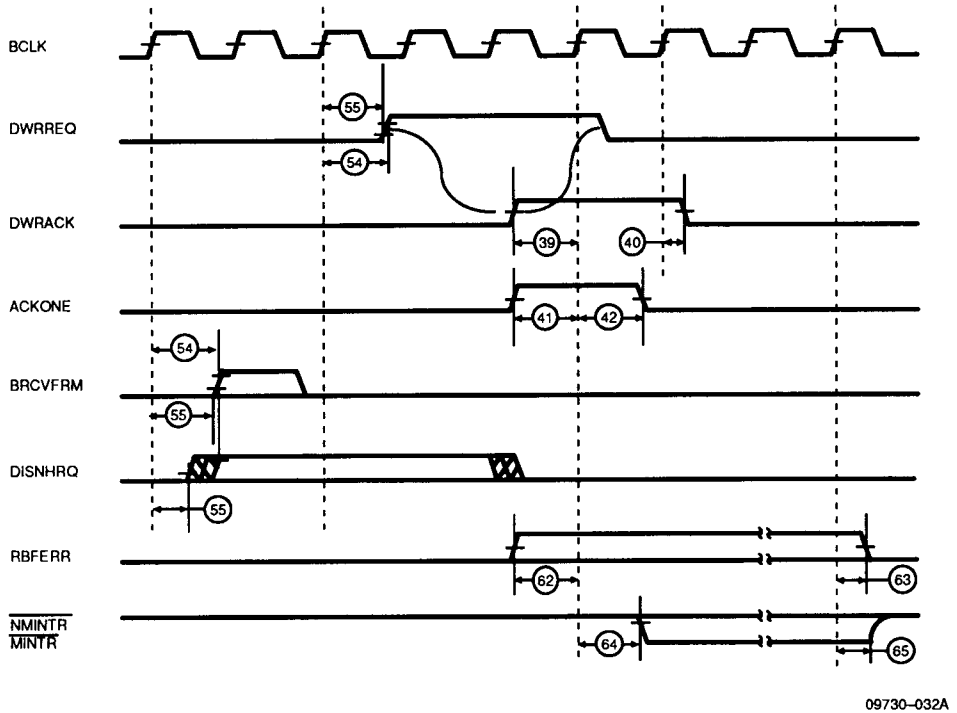
SWITCHING WAVEFORMS, (Continued)



DPC Frame Transmit Timing II

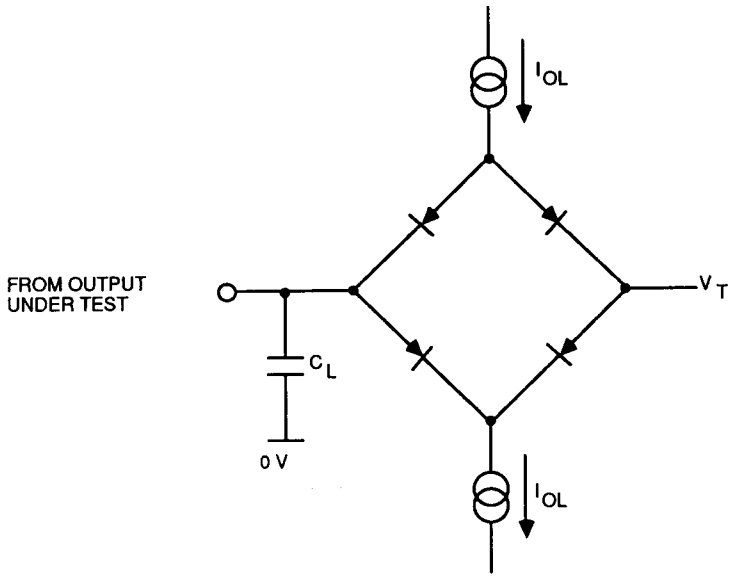


DPC Frame Receive Timing I



DPC Frame Receive Timing II

SWITCHING TEST CIRCUIT

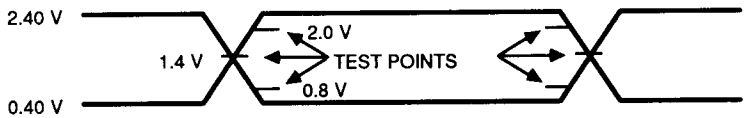


Note: C_L = 100 pF for pins NP_{0-15} $\overline{READY}$, $\overline{MINTR}$, $\overline{NMINTR}$, D_{0-31}, and DP_{0-3}. C_L = 50 pF for all other output pins.

09730-025A

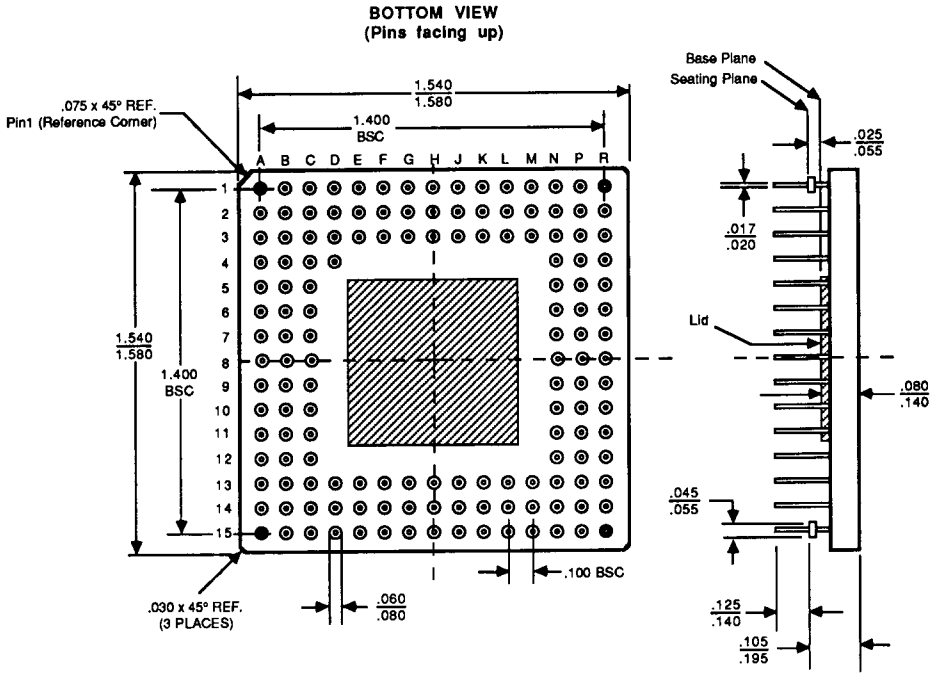
Standard Test Load

SWITCHING TEST WAVEFORM



Input/Output Waveform

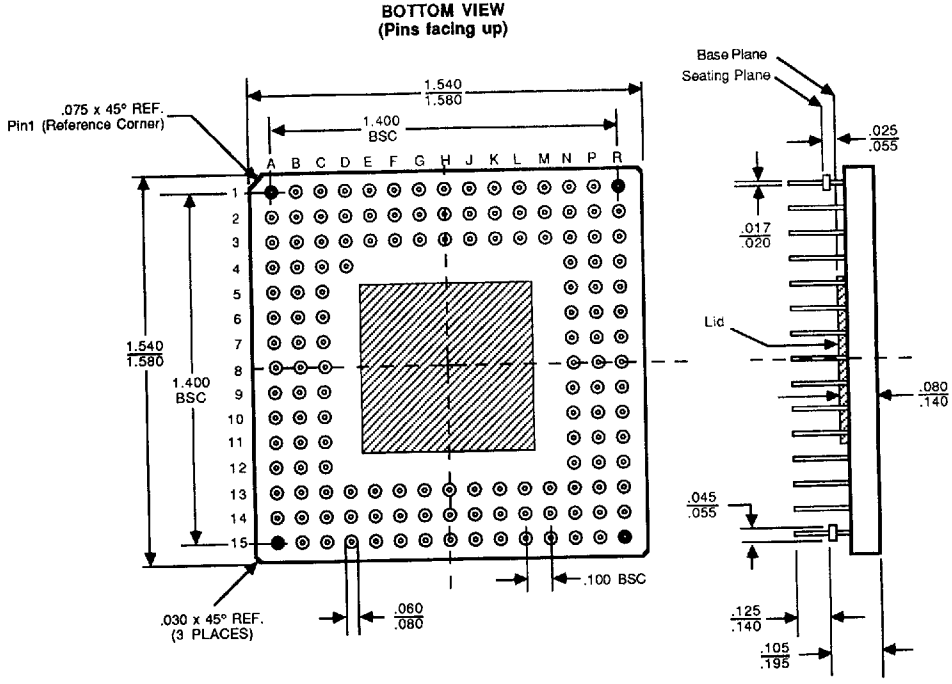
09730-026A


BOTTOM VIEW
(Pins facing up)
.075 x 45° REF.
Pin1 (Reference Corner)
1.540
1.580
1.400
BSC
A B C D E F G H J K L M N P R
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
1.540
1.580
1.400
BSC
.030 x 45° REF.
(3 PLACES)
.060
.080
.100 BSC
Base Plane
Seating Plane
.025
.055
.017
.020
Lid
.080
.140
.045
.055
.125
.140
.105
.195

#09691B

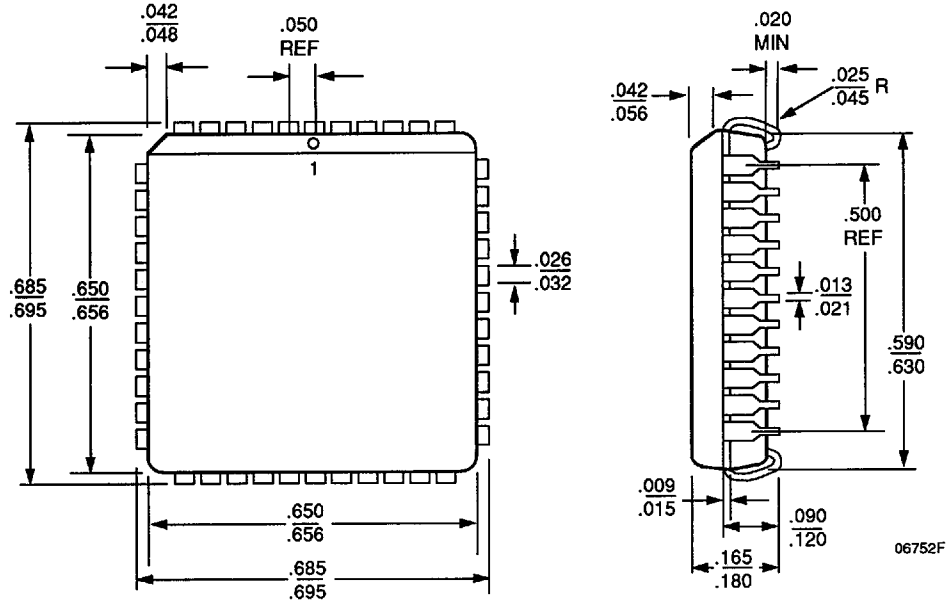
T-90-20

CGX145
145-Lead Pin Grid Array without Heat Sink

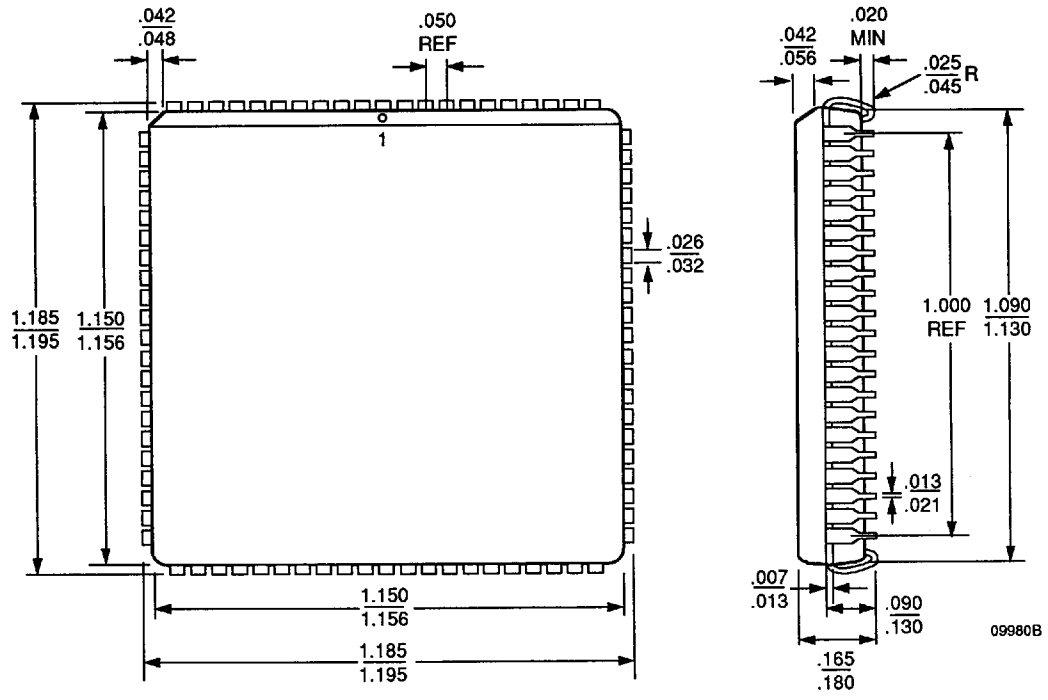


PID # 09691B

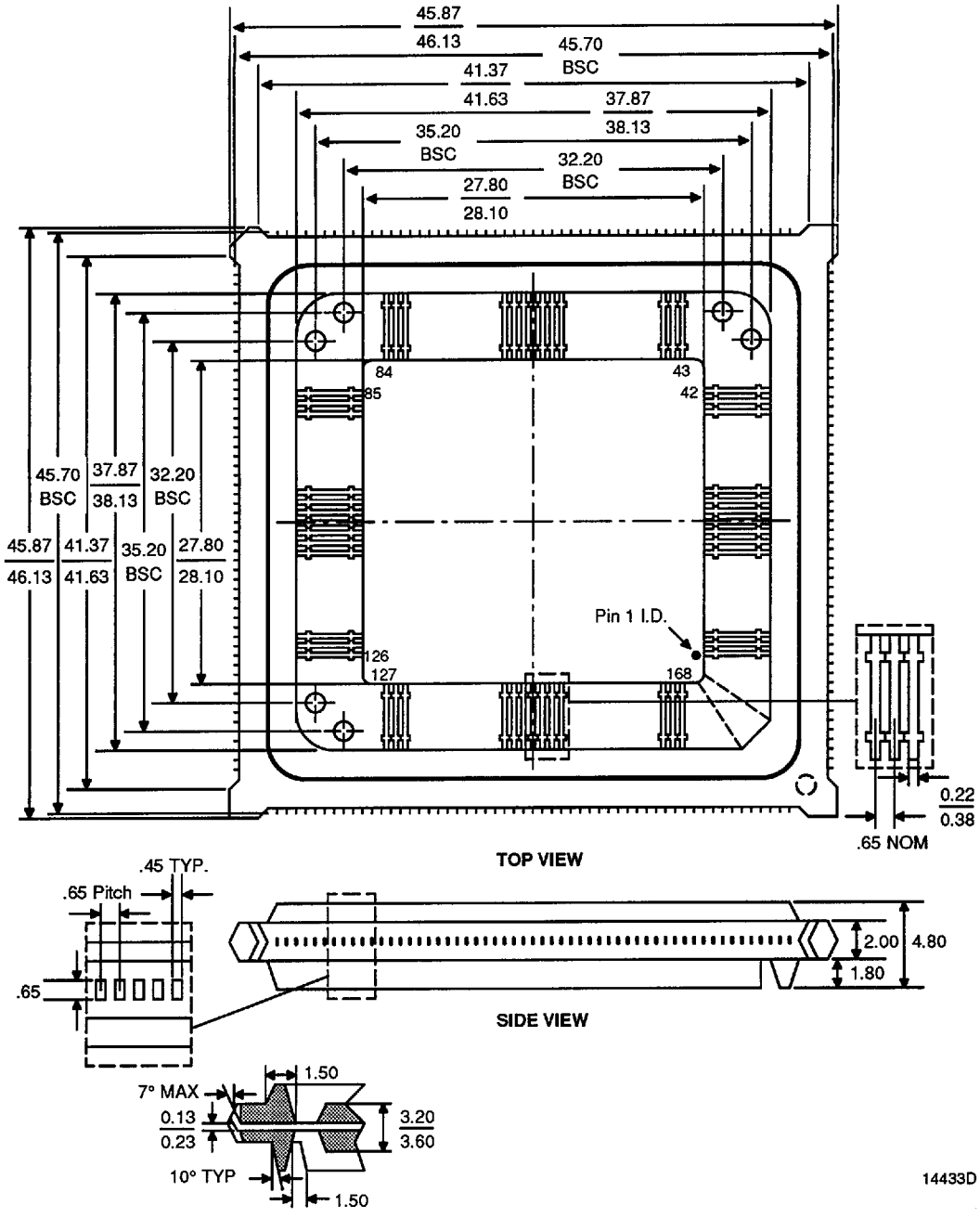
PL 044
44-Pin Plastic Leaded Chip Carrier



PL 084
84-Pin Plastic Leaded Chip Carrier

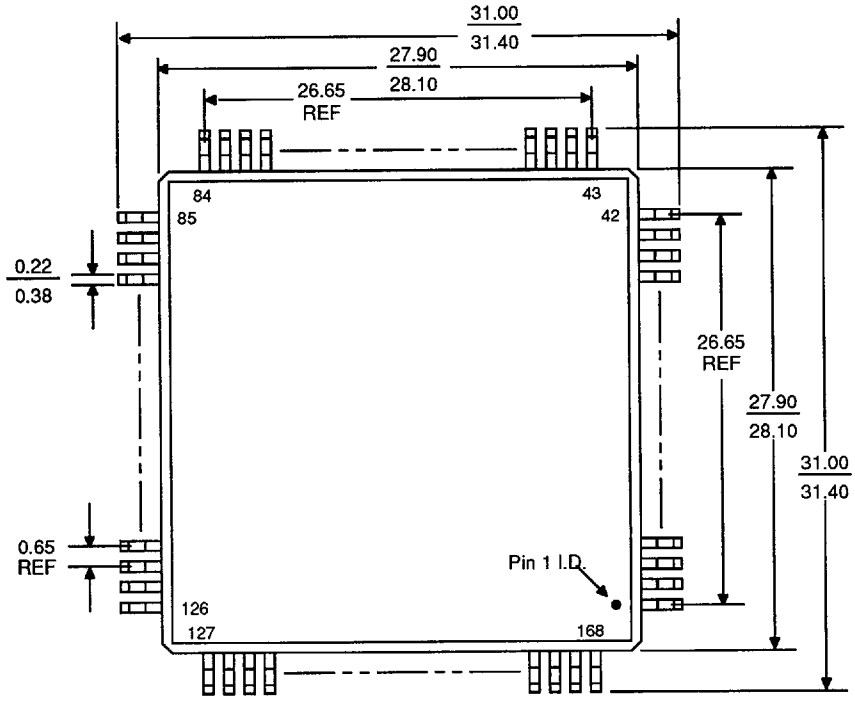


PQR168**
168-Pin Plastic Quad Flat Pack (Tape Pak)

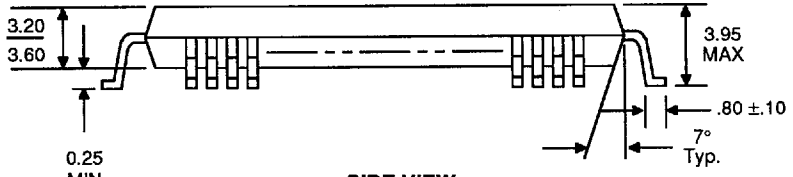


**Measured in Millimeters

PQJ168**
168-Pin Plastic Quad Flat Pack (Trimmed and Formed)



TOP VIEW



SIDE VIEW

**Measured in Millimeters