DP83231,DP83241,DP83251,DP83255

AN-727 A Guide to the Implementation of Physical Connection Management

Literature Number: SNOA196

A Guide to the Implementation of Physical Connection Management

National Semiconductor
Application Note 727
My Le, Michael Yip
February 1991

1.0 Introduction

The FDDI Station Management (SMT) standard provides the necessary control of an FDDI station (node) so that the node may work cooperatively as a part of an FDDI network. To effectively implement the functions required, SMT is divided into three entities, namely the Connection Management entity (CMT), the Ring Management entity (RMT) and also the Frame Based Services. The Connection Management is further divided into three entities, the Physical Connection Management (PCM), Configuration Element Management (CEM), and Entity Coordination Management (ECM).

The Physical Connection Management is an entity within Connection Management whose functions include:

- Initialization of the connection of neighboring ports pair where each port is comprised of one PMD entity and one PHY entity.
- Enforcement of port connection policies and withholding of unacceptable connections
- Testing of link confidence and monitoring of link quality between neighboring ports
- Detection of physical connection level faults between two ports and invocation of path test
- Support of maintenance line state
- Participate in the Trace action

For a general description of SMT, please consult the application note entitled ''SMT Simplified'', AN-726.

In this document, the operation of the Physical Connection Management entity is described along with a guide to the implementation of the PCM using the PLAYER™ device. In addition, one implementation of the Link Error Monitor using the PLAYER device is also discussed. For a more detailed description of PCM and other SMT entities, please consult the ANSI FDDI Station Management Standard.

2.0 PCM Functions Overview

Many instances of PCM can exist on a FDDI node and each instance of PCM controls one port in a node. For example, two separate instances of PCM exist in a Dual Attached Station (DAS) and only one instance of PCM exists in a Single Attached Station (SAS). Each PCM communicates with the ECM and CEM entities and directly controls the PHY layer device of a port.

One of the most important functions of PCM is to establish a connection between two ports. The connection process is achieved through a lock-step handshaking procedure. The handshaking procedure controlled by PCM is divided into three stages:

- Initialization sequence
- Signaling sequence
- Join sequence

The initialization sequence is used to indicate the beginning of the PCM handshaking process. It forces the neighboring PCM into a known state so that the two PCM state machines can run in a lock-step fashion.

Following the initialization sequence is the signaling sequence. The signaling sequence communicates information about the port and the node with the neighboring port. A Link Confidence Test (LCT) is also conducted during the signaling sequence to test the link quality between the two ports. If the link quality is not acceptable or the type of connection is not supported by the nodes then the connection will be withheld.

If the connection is not withheld during the signaling sequence, the PCM state machine can move onto the join sequence and establish the connection between the two neighboring ports.

In addition to establishing the connection, PCM also supports the Maintenance function and performs the Trace function. During the Maintenance function, PCM forces the PHY layer device to transmit a specified line state continuously. The Maintenance function allows SMT to force the PCM of the neighboring port into a known state manually and line faults may be traced when both end nodes are in known states and do not change state due to line noise.

The operation of PCM can be implemented by a state machine. The state machine is comprised of ten states: Off, Break, Trace, Connect, Next, Signal, Join, Verify, Active and Maintenance. The next section of this document describes the PCM State Machine.

3.0 Detailed PCM Description

3.1 CONNECTION SEQUENCE

The connection sequence starts with the reception of the PC__Start signal from the Entity Coordination Management. PC__Start indicates that the physical media is available for communication. The PC__Start signal causes the PCM state machine to enter the Break state which is the beginning of the initialization sequence.

In each sequence the two PCM state machines exchange a sequence of line states. The PCM state machine sends the line state to its neighboring PCM by directing the PHY layer devices to transmit a continuous stream of line state symbols. This line state is transmitted for a duration of time to ensure that the neighboring PCM receives the information. While transmitting, the PCM state machine also expects to receive a particular line state from the neighboring PCM. If things go as planned, the connection progresses through the three sequences and PCM signals the Configuration Element Management entity to include the connection into the token path.

RRD-B30M75/Printed in U. S. A.

The remainder of this section describes the detail operation of each sequence in the connection process.

Initialization Sequence

The Break state and the Connect state are used in the initialization sequence to start the connection.

The Break state is the entry point in the start of a PCM connection. In the Break state, a continuous stream of Quiet Symbols is transmitted to force the other end of the connection to break any existing connection and restart the connection sequence. The Break state is entered upon the reception of the PC__Start signal from ECM or other PCM states when external events force a reinitialization of the connection. Reinitialization is required if the Link Confidence Test fails, the expected line state is not received, a noise condition occurs for a period of time, or the neighboring port is forced to break the connection.

When Quiet line state or Halt line state is entered during Break state, the connection has been initialized successfully in the Break state and the PCM state machine can transition to the Connect state. The Connect state is used to synchronize the two ends of the connection to begin the signaling sequence. In the Connect state, a continuous stream of Halt symbols is transmitted for a sufficient amount of time for clock acquisition by the receiving PHY.

Signaling Sequence

The second stage of the connection sequence is the signaling sequence. The Next state and the Signal state is used in the signaling sequence to exchange port information with the neighboring PCM. Link Confidence Test and/or MAC Loop Back Test are also performed during the signaling sequence.

The PHY line states are used to signal bit information and provide handshaking during the signaling. During the signaling sequence, three line states are used. The Idle line state is used as a bit delimiter, the Halt line state is used to represent a logical one (set) and the Master line state is used to represent a logical zero (clear). If Quiet line state is entered during any part of the signaling sequence, the PCM state machine should make a transition to the Break state and restart the connection sequence again. During the signaling sequence, ten bits of information are exchanged with the neighboring PCM. Section 5.2 has a list of the format of the 10 bits of information.

The Initialization sequence leaves the PCM state machine in the Connect state while the PHY Port is transmitting Halt symbols. The reception of Halt line state in the Connect state causes the PCM state machine to transition to the Next state. The signaling sequence starts upon the first transition to the Next state. In the Next state, a continuous stream of Idle symbols is transmitted. The Next state is used to separate the bit signaling performed in the Signal state.

The reception of Idle line state in the Next state causes the PCM state machine to transition to the Signal state. In the Signal state, a continuous stream of Halt symbols or Halt-Quiet (Master) symbol pairs are transmitted. Therefore one bit of information is transferred each time when the Signal state is entered. The reception of Halt line state or Master line state in the Signal state will cause the PCM state machine to transition to the Next state again. The Next state and Signal state cycle repeats ten times to exchange all ten bits of information in the signaling sequence.

Join Sequence

The final stage of the connection sequence is the join sequence. The join sequence is comprised of three states running in sequence. The three states are Join state, Verify state and finally the Active stage. The join sequence is a unique sequence of transmitted symbol streams received as line states (HLS-MLS-ILS) that leads to an active connection. At the end of the join sequence, the PCM state machine signals the CEM entity and CEM will incorporate the connection into the token path.

The PCM state machine enters the Join state when the signaling sequence finishes exchanging the ten bits of information. After the tenth bit is transmitted, the PCM state machine enters the Next state. The reception of Idle line state in the Next state causes the PCM state machine to transition to the Join state. In the Join state, a continuous stream of Halt symbols is transmitted.

The reception of Halt line state in the Join state causes the PCM state machine to transition to the Verify state. In the Verify state, a continuous stream of Halt-Quiet (Master) symbol pairs is transmitted.

The last state of the connection sequence is the Active state. The reception of Master line state in the Verify state causes the PCM state machine to transition to the Active state. In the Active state, a continuous stream of Idle symbols is transmitted. Upon the reception of Idle line state in the Active state, the PCM state machine directs the PHY device into the Active Transmit Mode, which transmits the PDUs from the PHY device's request port.

Many timers are used to ensure the connection sequence proceeds in lock-step fashion on the two ports of the link. The timers used in PCM are listed in Section 5.3 together with a brief explanation of each timer. The state diagram of the PCM state machine and a list of PCM states are also presented in Section 5.1.

3.2 MAINTENANCE FUNCTION

The Maintenance state is used to perform the Maintenance Function. In the Maintenance state, the symbol stream specified by higher level management agent shall be forced. During the Maintenance state, the PCM state machine is insensitive to the received line state.

The Maintenance state is useful to ensure that the PHY and PMD devices can transmit all the symbols specified in the FDDI standard. The Maintenance state is also used to force the other end of the connection to a particular state manually (without going through the Connection Sequence).

3.3 TRACE SUPPORT

Trace support is needed to localize a Stuck Beacon condition. The Trace function is used to recover from the Stuck Beacon condition by propagating the Trace signal along the ring, causing all the stations and concentrators in the suspected fault domain to leave the ring and perform a Path Test. In the Trace state, a continuous stream of Halt-Quiet (Master) symbol pairs is transmitted to the upstream station.

Two possible sequences of events can cause the PCM state machine to enter the Trace state. In the first case, the reception of Master (or Trace) line state in the Active state causes the PCM state machine to send the Trace__Prop signal to ECM. ECM will direct the appropriate PCM state machine to transition to the Trace state by sending the PC__Trace signal to that PCM. Upon the reception of the PC__Trace signal, a continuous stream of Halt-Quiet (Master) symbol pairs is transmitted.

The second situation that causes PCM state machine to enter the Trace state is a bit more involved. When the Ring Management entity detects a Stuck Beacon condition, RMT sends the Trace_Prop signal to the ECM. Like the first case, the ECM entity will direct the appropriated PCM state machine to transition to the Trace state.

3.4 Link Confidence Test

The Link Confidence Test (LCT) tests the link quality before the link is inserted into the token path. If the link quality is not adequate for ring operation then LCT will fail. Failure of LCT will cause PCM to return to the Break state and the connection is reinitialized. As a result, the LCT and the whole connection sequence will be repeated until the LCT is passed. The LCT is intended to detect major link problems and not to determine the exact Link Error Rate.

The Link Confidence Test is performed after bit 6 is transmitted in the Signaling sequence. During the duration of the test, either Idle symbols or valid PDUs are transmitted. The reception of Master line state signifies the successful completion of the LCT, while the reception of Halt line state signifies the failure of the LCT. If Quiet line state is entered during the LCT, the test is aborted and PCM returns to the Break state.

The duration of the LCT and the type of LCT is determined by the bit signaling process in the signaling sequence. LCT is defined to have four durations, namely LC_Short (50 ms), LC_Medium (500 ms), LC_Long (5 sec) and LC_Extended (50 sec). When a port is started the first time, it requests the shortest test duration. Every time the LCT fails, the duration is increased up to a maximum duration of LC_Extended. If the two ports on the link request different LCT duration in the Signaling sequence, the longer LCT duration is used to ensure a better measurement of link quality.

The LCT can be performed using one of the following methods:

- Transmit Idle symbols and count link errors. The error measurement is taken from the PHY device.
- Transmit valid PDUs and count link errors. Again, the error measurement is taken from the PHY layer device.
- Transmit valid PDUs and count Frame Check Sequence (FCS) errors from the MAC frames. In this method, the MAC layer device is used to source the PDUs and also take error measurements.
- The PHY will retransmit what is received and the transmitter will check for link errors. This method requires at least one MAC device connected to the link to send PDUs. The error measurement is taken from the PHY layer device.

Because multiple levels of the LCT can be performed, a minimum capability of transmitting Idle symbols and counting link errors is required for each port.

3.5 LINK ERROR MONITOR

The Link Error Monitor (LEM) continuously examines the link error rate (LER) of an Active connection. Its function complements the LCT to ensure that the link quality is adequate for ring operation at all times. When the LER exceeds the cutoff value, the connection is flagged as faulty and shall be removed from the token path.

The LEM and LER are based on the link error count from the PHY level device. However, the implementation of LEM and LER is not specified in the FDDI SMT Standard. One way of calculating LER is to keep a running time average of link error events occurred in a given time period. See Table I.

TABLE I. Link Error Rate (LER) Calculation Example

Time (in ms)	LEC in	Starting Time	Ending Time	Total Time (in ms)	Total LEC	LER (in Bit/Sec)
T+100	0	T	T+100	100	0	$<8.00 \times 10^{-8}$
T+200	1	T+100	T+200	200	1	4.00×10^{-8}
T+300	0	T+100	T+300	300	1	2.67×10^{-8}
T+400	0	T+200	T+400	300	1	2.67×10^{-8}
T+500	0	T+300	T+500	300	0	$<2.67 \times 10^{-8}$
T+600	2	T+400	T+600	300	2	5.33×10^{-8}
T+700	1	T+500	T+700	300	3	8.00×10^{-8}
T+800	0	T+600	T+800	300	3	8.00×10^{-8}
T+900	0	T+700	T+900	300	1	2.67×10^{-8}

Notes: The Time field is in ms and the measurement is started from certain time and ended 100 ms later.

The Link Error Count (LEC) field measures the number of Link Error occurring in the 100 ms interval.

The Starting and Ending Time is the starting and ending period for the LER calculation.

LER is calculated by the formula:

$$\text{LER} = \frac{\text{Total LEC}}{\text{Total Time (seconds)} \times 125 \times 10^6}$$

Note that LER cannot be zero even if Total LEC is zero and the LEM can only indicate the LER is less than when Total LEC is one.

4.0 PLAYER Device

The PLAYER device was designed with SMT in mind, and therefore the PLAYER device offers a larger number of registers to be used by the PCM software. These registers can be divided into four groups based on their functions. The groups are general control and system interface registers, line state reception and symbol generation registers and event counter registers.

The general control and system interface registers provide control to different types of optical transmitters and receivers, control of different types of transmission and reception and also control of interrupt generation for the system. The line state reception and symbol generation registers allow the transmission of different line states and the reporting of line state reception (with or without interrupt generation). The event counter registers consist of three sets of registers, one set keeps track of noise events, one set for link error monitoring and another set for timing of the incoming line state.

The following sections discuss the PLAYER registers in detail and also some ideas on how to implement PCM using the PLAYER device. For a complete list of PLAYER registers, please refer to the DP83251/55 PLAYER device datasheet.

4.1 PLAYER REGISTERS

The PLAYER device includes many registers which ease the implementation of PCM. This section provides an explanation of the registers which are used to implement PCM and how to use those registers. Consult the PLAYER datasheet for the address of those registers. Note that not all the functions in each registers are discussed in this section.

MODE Register

The Mode Register (MR) controls the mode of operation of the PLAYER device. MR enables and disables the PLAYER device, which is a very important function! It also sets the transmission level of the Quiet symbols to either low level or high level so that the PLAYER device can interface with different types of optical transmitters without extra external logic.

Interrupt Condition Register

The Interrupt Condition Register (ICR) records the occurrence of an interrupt condition. It is set by the hardware and can be clear by the software. Each bit in the ICR represents an interrupt condition, the interrupt conditions are:

- PHY Request Interface Parity Error. This error occurs when there is a parity error at the PHY request interface and the parity checking is enabled by setting the PHY Request Data Parity Enable bit in the Current Transmit State Register.
- Control Bus Data Parity Error. This error occurs when there is a parity error in the incoming data from the Control Bus and the parity checking is enabled by using the Control Bus Parity Enable pin.
- Control Bus Write Command Reject. This error occurs when the software is trying to write into one of the read-only registers in the PLAYER device.
- Conditional Write Inhibit Condition. This bit is set to one, when the software tries to write to a conditional register when that register is updated internally. This feature protects the system from accidentally erasing an exception condition.
- Link Error Monitor Threshold Reached. This bit is set to one when the internal 8-bit Link Error Counter reaches zero. It is used to implement the alarm limit and the cutoff limit of the LEM.
- Receive Condition A. This bit is set to one by the PLAYER when the Receive Condition A occurs. See also the Receive Condition Register A and Receive Condition Mask Register A.
- Receive Condition B. This bit is set to one by the PLAYER when the Receive Condition B occurs. See also the Receive Condition Register B and Receive Condition Mask Register B.
- User Definable Interrupt. The bit is set when one or both of the Sense Bits in the User Definable Register is set to one. In order to clear this bit, both the Sense Bits must be set to zero.

An interrupt is not generated unless the bit(s) in the ICR is set and the corresponding bit(s) in the Interrupt Condition Mask Register (ICMR) is also set.

Interrupt Condition Mask Register

The Interrupt Condition Mask Register (ICMR) is used to determine which event in the Interrupt Condition Register can generate an interrupt.

Interrupt Condition Comparison Register

The Interrupt Condition Comparison Register (ICCR) ensures that any changes in the Interrupt Condition Register (ICR) is recorded while it is being read by or written to the Control Bus Interface.

This register can also be used to reset all bits in the Interrupt Condition Register. This task is accomplished by setting all bits in ICCR to one and reset all the bits in ICR to zero. If the interrupt condition is not cleared before resetting bits in ICR, those bits will be set again by the PLAYER device.

Current Transmit State Register

The Current Transmit State Register (CTSR) can be used to program the following functions:

- Generate and transmit symbols
- Control Injection Mode
- Enable the Smoother in the Transmit block
- Enable parity at the PHY Request Interface

The symbol generation and transmission function is used by the PCM state machine to send out FDDI control symbols. The symbols can be generated by setting the three least significant bits in CTSR. Table II shows the Transmit Mode and the corresponding bit assignments.

TABLE II. Transmit Mode Bit Assignment

Transmit Mode	Bit(2)	Bit(1)	Bit(0)
Active	0	0	0
Idle	0	0	1
Master	1	0	0
Halt	1	0	1
Quiet	1	1	0

When the Transmit Mode is set to Idle mode or Halt mode, the PLAYER generates and transmits the Idle symbols or Halt symbols respectively. When it is set to Master mode, the PLAYER transmits the Halt-Quiet symbol pair. In the Quiet Transmit Mode, a stream of Quiet symbols is transmitted using the Quiet Transmit Level programmed in the Mode Register. During the Active Transmit Mode, the transmit block repeats the request data onto its outputs, therefore, Idle symbols and PDUs can be transmitted.

Current Transmit State Comparison Register

The Current Transmit State Comparison Register (CTSCR) ensures that any changes in the Current Transmit State Register (CTSR) are recorded while it is being read or written to by the Control Bus Interface. This register can also be used to reset all bits in the Current Transmit State Register.

Current Receive State Register

Line states being received by the PLAYER device are reported via the Current Receive State Register (CRSR). At the reception of the new line state, information about the previous line state is cleared. This register is different from the two Receive Condition Registers in that the Receive Condition Registers create a historical record of all the line states received whereas the CRSR only reflects the current line state.

Table III shows the line states and the decoded bit assignments.

TABLE III. Line State Bit Assignment

Line State	Bit(2)	Bit(1)	Bit(0)
Active	0	0	0
Idle	0	0	1
No Signal	0	1	0
Master	1	0	0
Halt	1	0	1
Quiet	1	1	0
Noise	1	1	1

The No Signal Detected condition and the reception of Quiet line state condition are very similar. In Quiet line state, a stream of Quiet symbols is received. However, the No Signal Detected condition is reported when the Signal Detect pin (TTLSD) has been deasserted. When the TTLSD pin is deasserted, the receiver probably also receives a stream of Quiet symbols.

Note that CRSR is a read-only register and is often used in a polling application since it cannot be used to generate interrupts.

Receive Condition Register A and Receive Condition Register B

The Receive Condition Registers A and B (RCRA and RCRB) are used to maintain a historical record of the line states received by the PLAYER device.

When a new line state is received, the bit corresponding to the line state is set without clearing any other bits. As a result, line states previously received are also maintained. Although all line states received are being recorded, the registers do not keep track of the sequence of line state reception.

Since the function of the Receive Condition Registers is to record line state changes, not the current line state, it is possible to completely clear the registers by the code sequence in Table IV. Note that RCRA is zero at the end of the code sequence even when a Halt line state is being received.

TABLE IV. Clearing the Receive Condition Register

Time	CRSR	RCRA	Action
T0	QLS	0000 0010	None
T1	QLS	0000 0010	None
T2	HLS	0100 0110	None
T3	HLS	0100 0110	Read RCRA
T4	HLS	0100 0110	Write RCRA = 0
T5	HLS	0000 0000	None

Receive Condition Mask Register A and Receive Condition Mask Register B

The PLAYER can be programmed to generate an interrupt when a line state of interest to the PCM State Machine is received. When a bit in the Receive Condition Register is set by the PLAYER device and the corresponding bit in the Receive Condition Mask Register (RCMRA or RCMRB) is also set, then an interrupt condition is generated and is registered in the Interrupt Condition Register. However, an interrupt is not generated unless the corresponding bits (Bit5 and Bit6) are also set.

Receive Condition Comparison Register A and Receive Condition Comparison Register B

The Receive Condition Comparison Registers (RCCRA and RCCRB) ensures that any changes in the Receive Condition Registers is recorded while they are being read or written by the Control Bus Interface.

These registers can also be used to reset all bits in the Receive Condition Registers. This task is accomplished by setting all bits in RCCRs to one and writing zero to the Receive Condition Registers.

Noise Threshold Register and Noise Prescale Threshold Register

The Noise Threshold Register (NTR) and Noise Prescale Threshold Register (NPTR) are used to set the threshold value of the internal 15-bit noise counter. The noise counter counts the duration that the PLAYER device receives noise.

The Noise Threshold Register is a 7-bit register. It forms the most significant bits of the noise threshold value. The Noise Prescale Threshold Register is an 8-bit register. It forms the least significant bits of the noise threshold value. Therefore, the noise counter takes

$$((NPTR + 1) \times (NTR + 1)) \times 80 \text{ ns}$$

to reach zero. For example, if NPTR is 1100 0111 (199),

then NTR is set to 0110 0011 (99) for a threshold value of 1.6 ms. Note that 1.6 ms is the default value of the NS__Max timer used in PCM. Also note that the noise counter can be set at the maximum threshold value of 2.62 ms.

When the noise threshold is reached, the Noise Threshold (Bit5) in the Receive Condition Register is set to one, which can in turn generate an interrupt.

Current Noise Count Register and Current Noise Prescale Count Register

The Current Noise Count Register (CNCR) and Current Noise Prescale Count Register (CNPCR) takes a snap shot of the internal noise counter. CNCR reports the most significant 7 bits of the noise counter, whereas the CNPCR is for the least significant 8 bits. Both registers are read-only registers.

Link Error Threshold Register

The Link Error Threshold Register (LETR) contains the starting value of the internal Link Error Monitor counter. The LEM counter is an 8-bit down-counter which decrements if link errors are detected.

The LETR value is loaded into the LEM counter when a value is written to the LETR or when the internal LEM counter reaches zero. When the internal LEM counter reaches zero, an interrupt condition is generated. The interrupt condition is registered in the Interrupt Condition Register. If the corresponding bit (Bit4) in the Interrupt Condition Mask Register is also set, an interrupt is generated.

Current Link Error Count Register

The Current Link Error Count Register (CLECR) serves a function similar to the Current Noise Count Registers. The CLECR takes a snap shot of the internal LEM counter, thus allowing the control process to read the LEM counter without interrupting the LEM counter. The CLECR can be used to calculate the Link Error Rate and it is also needed in the Link Confidence Test during the PCM connection sequence.

State Threshold Register and State Prescale Threshold Register

The State Threshold Register (STR) and the State Prescale Threshold Register (CPTR) are used to set the threshold value of the internal 15-bit state counter. The state counter counts the duration that the PLAYER device receives a line state.

The State threshold Register is a 7-bit register. It forms the most significant bits of the state counter threshold value. The State Prescale Threshold Register is an 8-bit register. It forms the least significant bits of the state counter threshold value. Like the Noise Threshold Registers, the STR and SPTR together can specify a threshold value of 2.62 ms. When the internal state counter reaches zero, the State Threshold (Bit1) is set to one, therefore, an interrupt can be generated.

The internal state counter can be used to keep track of the length of time a particular line state is required to be transmitted or received before the PCM state machine can take appropriate actions. For example, while the PCM state machine is in the Next state, the State Counter can be set at a threshold of 1.6 ms to ensure that the state machine has sent a sufficient number of line state symbols in the Connection state before it moves to the Next state.

Current State Count Register and Current State Prescale Count Register

The Current State Count Register (CSCR) and the Current State Prescale Count Register (CSPCR) take a snap shot of the internal state counter. CSCR reports the most significant 7 bits of the state counter, whereas the CNPCR is for the least significant 8 bits. Both registers are read-only registers.

4.2 POLLING TECHNIQUE

Many techniques can be used by the PCM state machine software to monitor the line state information and the polling technique is one of them.

A sample of a polling algorithm is shown in *Figure 1*. In the example, the PCM state machine just moved to the Next state. In the Next state, the PCM state machine needs to transmit Idle symbols, poll for Quiet line state or Idle line state, and transmit a ''bit'' signal if idle line state is received.

```
01 CTSR = ITM
02 Loop
03  RxLS = CRSR
04  IF RxLS = QLS
     THEN BreakCondition = TRUE, Done = TRUE
05  IF RxLS = SILS
     THEN LSFlag = TRUE, Done = TRUE
06  IF TPC > T.Out
     THEN BreakCondition = TRUE, Done = TRUE
07 End Loop if Done = TRUE
08 Wait (TL.Min)
09 IF BreakCondition = TRUE
    THEN DO BreakAction
10 IF LSFlag = TRUE
    THEN TxLS = RCode(n), CTSR = TxLS
```

FIGURE 1. Polling Technique Example Code

Let's examine the code in more detail. Line 01 instructs the PLAYER device to transmit Idle symbols. Line 02 is the beginning of the polling loop which ends at line 07.

In the polling loop, line 03 polls the CRSR register and stores the received line state in the variable RxLS. If received line state is Quiet line state, line 04 sets the BreakCondition variable to indicate the PCM state machine needs to return to the Break state. If the received line state is Super Idle line state, it sets the variable LSFlag, indicating that the correct line state is received. Super Idle line state is tested instead of Idle line state because it is required to wait for at least 12 Idle symbols in the Next state before any action. Line 06 checks for the time out condition to ensure the state machine does not get stuck in the Next state waiting for Idle symbols. The polling loop is repeated until one of the conditions is met.

After existing the polling loop, the PCM state machine has to wait for 30 ms (TL__Min) before starting the next action (Line 08). It is used to make sure that the PHY device on the other side has enough time to recognize the line state symbols. If the BreakCondition is TRUE, line 09 will transit to the Break state by executing a subroutine BreakAction. If everything is fine and the expected line state is received, the PCM Psuedo Code machine determines the next transmission mode and directs the PCM state machine to transmit the new line state symbols.

During the polling loop, the PCM software needs to monitor the Current Receive State Register very frequently so that line state information will not be missed. Failure to recognize all the changes of line state can cause failure in the connection sequence and reinitialization of the connection.

4.3 INTERRUPT TECHNIQUE

The interrupt technique can also be used in implementing the PCM software. This section explains a simple sample of a portion of the PCM software using the interrupt capability built in to the PLAYER device.

The code segment shown in *Figure 2* is part of an interrupt handler that responds to the line state reception during the signaling sequence. It serves the same purpose as the code shown in the previous section.

```
01  RCMRA = 0, RMMRB = 0
02  tmp = ICR
03  tmp = tmp AND 1001 0000
04  ICR = tmp
05  RxLS = CRSR
06  IF RxLS = QLS
       THEN DO BreakAction, RETURN
07  IF RxLS = ILS
08     THEN   TxLS = RCode(n),
09            tmp = RCRA, RCRA = 0,
10            tmp = RCRB, RCRB = 0,
11            RxLS = CRSR,
12            RCRA = RCRARxMsk [RxLS],
13            RCRB = RCRBRxMsk [RxLS],
14            RCMRA = 0010 1111,
15            RCMRB = 0000 0000,
16            CTSR = TxLS
17  RETURN
```

FIGURE 2. Interrupt Technique Example Code

Line 01 is first line of this part of the interrupt handler. The interrupt handler has to examine the device and condition that generates the interrupt before executing the code.

Lines 01 to 04 stop the PLAYER device from generating the interrupt. It first stops the interrupt condition by clearing both RCMRA and RCMRB. Then, it clears the Receive Condition A (Bit5) and the Receive Condition B (Bit6) in the Interrupt Condition Register. Note that the ICR is a conditional write register, as a result, it is read in line 02 before a value is written to it.

Lines 05 to 07 decide what to do with the received line state. If the current line state is Quiet line state, BreakAction is executed and this section of the interrupt handler is completed. However, if the received line state is Idle line state, the interrupt handler needs to prepare the PLAYER for the reception of the next line state and also transmits the next line state.

It first runs the PCM Psuedo Code machine by calling the subroutine RCode(n), where n is the nth bit during the signaling sequence. Lines 09 and 10, prepare the reception of the next line state by clearing the registers RCRA and RCRB. However, clearing all the line states in the RCRA and RCRB register means that the current line state information is also cleared from the two registers. As a result, the line state that PCM is expecting may also be erased. Lines 11 to 13, prevent such a condition by writing the current line state information back to RCRA or RCRB.

At this point, the PLAYER device is ready to receive new line states. Line 14 sets up the mask for RCRA so that when the expected line state is received, an interrupt can be generated. The value 0010 1111 written to RCMRA in this example allows the PLAYER to watch for Halt line state, Master line state, Quiet line state and also the Noise condition. All these are required in the Next state. Line 15 clears the mask for RCRB. This line is not necessary since RCMRB is cleared at the beginning of the code. It is only included as a reminder for other PCM states.

The next line state is finally transmitted in line 16. After the new line state is transmitted, this code segment ends and returns to the calling process.

4.4 OTHER TECHNIQUES

Interrupt and polling are by far the most popular techniques used to implement PCM software. However, implementation of the PCM software is not limited by the two techniques. One other approach is to use the interrupt technique together with an event queue schedular.

The mix design that uses the interrupt technique and the event queue provides less interrupt handling time and decouples the interrupt handling code from the PCM software. When an event or line state occurs and an interrupt is generated, the interrupt handler unmasks the interrupt condition and enqueues the line state event into the event queue. Events in the event queue are taken care of one after another outside of the interrupt handling routines.

Many more possible designs can be used to implement PCM with the PLAYER device. The implementations depend on the underlying hardware platform and cannot be covered by this paper.

5.0 Appendix

This section provides explanations and definitions of terms needed to help understand PCM. However, the ANSI SMT document shall be used as the standard reference. The State diagram from the ANSI SMT document is reprinted in *Figures 3* and *4*.

5.1 PCM STATES

The PCM State Machine is comprised of ten states. Section 4.1 explains the meaning and functions of each PCM state.

PC0: Off State

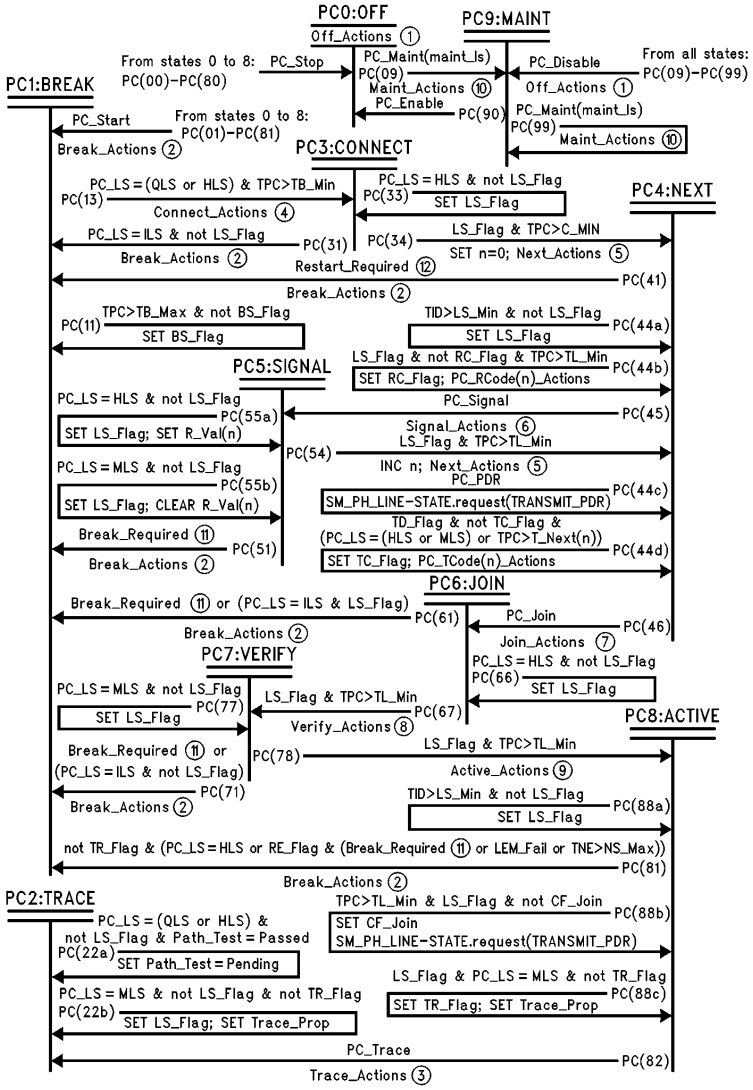
The Off state is the initial state of the PCM State Machine. The PCM returns to this state upon the reception of the PC_Stop signal from the Entity Coordination Management Entity (ECM).

In the Off state, the PMD optical transmitter is optionally disabled and the PHY device is required to transmit Quiet symbols.

PC1: Break State

The Break state is the entry point of the PCM connection sequence. The Break state is entered upon the reception of the PC_Start signal from ECM. The Break state is also entered from any other state when the connection sequence cannot be completed and a reinitialization is required.

In the Break state, the PMD optical transmitter is disabled and the PHY device is required to transmit Quiet symbols. A few variables are also cleared and intialized during the Break state.



TL/F/11081–1

FIGURE 3. PCM State Diagram

Physical Connection Management Footnotes.

1. Off__Actions:
   SM__PM__CONTROL.request(Transmit__Disable)*
   SM__PH__Line-State.request(TRANSMIT__QUIET)
   CLEAR CF__Loop
   CLEAR CF__Join
   CLEAR BS__Flag
2. Break__Actions:
   SM__PM__CONTROL.request(Transmit__Disable)*
   SM__PH__Line-State.request(TRANSMIT__QUIET)
   CLEAR CF__Loop
   CLEAR CF__Join
   CLEAR BS__Flag
   PC__Mode = N
3. Trace__Actions:
   CLEAR LS__Flag
   SM__PH__Line-State.request(TRANSMIT__MASTER)
4. Connect__Actions:
   CLEAR LS__Flag
   CLEAR BS__Flag
   SM__PM__CONTROL.request(Transmit__Enable)*
   SM__PH__Line-State.request(TRANSMIT__HALT)
5. Next__Actions:
   CLEAR LS__Flag, RC__Flag, TC__Flag, TD__Flag
   SM__PH__Line-State.request(TRANSMIT__IDLE)
6. Signal__Actions:
   CLEAR LS__Flag
   IF T__Val(n)
      THEN SM__PH__Line-State.request(TRANSMIT__HALT)
      ELSE SM__PH__Line-State.request(TRANSMIT__MASTER)
7. Join__Actions:
   CLEAR LS__Flag
   SM__PH__Line-state.request(TRANSMIT__HALT)
8. Verify__Actions:
   CLEAR LS__Flag
   SM__PH__Line-State.request(TRANSMIT__MASTER)
9. Active__Actions:
   CLEAR LS__Flag, TR__Flag
   SM__PH__Line-State.request(TRANSMIT__IDLE)
10. Maint__Actions:
    IF maintls = QUIET
       THEN SM__PM__CONTROL.request(Transmit__Disable)*
       ELSE SM__PM__CONTROL.request(Transmit__Enable)*
    SM__PH__Line-State.request(TRANSMIT__maintls)
11. Break__Required:
    PC__LS=QLS or (not LS__Flag & TPC>T__Out)
    Optionally, Break__Required may include the condition TNE>NS__Max:
    PC__LS=QLS or (not LS__Flag & TPC>T__Out) or TNE>NS__Max
12. Restart__Required:
    PC__LS=QLS or (not LS__Flag & n≠0 & TPC>T__Out)
    Optionally, Restart__Required may include the condition TNE>NS__Max:
    PC__LS=QLS or (not LS__Flag & n≠0 & TPC>T__Out) or TNE>NS__Max

13. Transitions effected by the Break__Required or Restart__Required conditions shall take precedence over other transitions. The reception of QLS while in states 4 to 8 shall cause the PCM to transition to the Break State within PC__React time.

14. RESET TPC on every transition. This includes transitions where the destination state is the same as the originating state.

*This primitive is not required for all PMD implementations.

FIGURE 4. PCM State Diagram Footnote

Upon the reception of Quiet line state or Halt line state, the PCM state machine leaves the Break state and enters the Connect state. If the state machine is stuck in the Break state for a long time (TB__Max), the state machine sets the BS__Flag so that other management agents can examine the condition and takes the appropriate action.

PC2: Trace

The Trace state is used to localize the fault domain of a Stuck Beacon condition where the ring cannot recover from its Beacon state. The state machine enters the Trace state when receiving the PC__Trace signal from ECM while in the Active state.

During the Trace state, the PHY entity transmits a stream of Halt-Quiet (Master) symbol pairs.

The only way to leave the Trace state is receiving the PC__Start or PC__Off signals from ECM.

PC3: Connection State

The Connection state is used to synchronize the two ends of the connection to begin the signaling sequence. It is also used for clock recovery since the Break state does not transmit any clocking information through the optical transmitter.

In the Connect state, the PMD level optical transmitter is enabled and a continuous stream of Halt symbols is transmitted.

Upon the reception of Halt line state, the state machine leaves the Connect state to the Next state. If Idle line state is received before Halt line state, then the connection is not synchronized and the state machine transmits to the Break state to restart the connection sequence.

PC4: Next State

The Next state is one of the two states used in the signaling sequence. The main purpose of the Next state is to separate the ''bit'' signaling performed in the Signal state. The Next state is also used to transmit PDUs while MAC Local Loop or Link Confidence Test is performed. The PCM Psuedo Code machine is also started in the Next state.

On initial entry into the Next state, a continuous stream of Idle symbols is transmitted. While in the Next state, either a continuous stream of Idle symbols or PDU symbol stream is transmitted.

The Next state terminates and the state machine transits to the Signal state upon the reception of Halt or Master lilne state or when the PC__Signal signal is received from the Psuedo Code machine. The state machine transit to the Break state upon the reception of Quiet line state.

PC5: Signal State

The Signal state is one of the two states used for the signaling sequence. In the Signal state, individual bits of information are communicated across the connection by transmitting either Halt symbols or Halt-Quiet (Master) symbols pair. Transmitting the Halt symbols is equated to the transmission of a logical one and the transmission of the Master symbols pairs is a logical zero.

Once each individual bit has been transmitted and received, the state machine moves to the Next state before returning to the Signal state for the next transmission. Thus the Next state is used as the bit delimiter between two signaling bits.

When all signaled bits have been transmitted and received, the Signaling Sequence ends.

PC6: Join State

The Join state is the first of three states in the join sequence that leads to an active connection. The join sequence assures that both ends of a connection enter the Active state together at the completion of the sequence.

The Join state is entered upon the completion of the signaling sequence when the PC__Join signal is issued from the Psuedo Code machine. In Join state, a continuous stream of Halt symbols is transmitted.

PC7: Verify State

The Verify is the second of three states in the join sequence. The Verify state is entered when Halt line state is received when the state machine is in the Join state.

A continuous stream of Halt-Quiet (Master) symbols pairs is transmitted during the Verify state.

PC8: Active State

The Active state is the last of three states in the join sequence. In this state, the port is incorporated into the token path.

On initial entry into the Active state, a continuous stream of Idle symbols is transmitted. Upon the reception of Idle line state during the Active state, the PHY device is allowed to enter the Active Transmit Mode and PDUs presented to the PHY Port Request interface can be transmitted.

In addition to the normal break conditions, when Halt line state is received in place of Idle line state, the connection is not synchronized and a reinitialization of the connection is required. If the Link Error Rate is too high, the connection also needs to be reinitialized upon entering the Active state.

PC9: Maintenance State

The Maintenance state is used to perform the Maintenance function. The Maintenance state is entered upon the reception of the PC__Maint or PC__Disable signal from other management agency.

5.2 PCM PSEUDO CODE

The PCM Pseudo Code provides and processes the information that is sent between the neighboring PCMs during the signaling sequence. As mentioned in the previous sections, the information is communicated via the bit signaling technique whereas a bit value is represented by a stream of line state symbols. The Halt symbols are used to represent a logical one and the Master symbols pairs for a logical zero.

This section explains the meaning of the ten bits of information communicated during the signaling sequence.

Bit(0): Escape Bit

Bit(0) is called the Escape Bit. It's value is zero for SMT Version 6.2. The setting of Bit(0) is reserved for future assignment by the standard.

Bit(1,2): PC__Type

Bit(1) and Bit(2) together state the PC__Type of this port. The four PC__Types defined in the FDDI standard are encoded as shown in Table V.

TABLE V. PC__Type Encoding

PC__Type	Bit(1)	Bit(2)
PHY__A	0	0
PHY__B	0	1
PHY__S	1	0
PHY__M	1	1

The PC__Type is communicated during the signaling sequence so that the connection policy can be enforced before the station is inserted into the token path.

Bit(3): Port Compatible

Bit(3) is set to one if the two ports on the link are compatible which means that this connection is allowed and supported by both stations. Connections that are not allowed are withheld until Bit(9) is received and a PC__Start signal is generated to reinitialize the connection. All the connections are allowed in the standard except a M−M connection, however certain types of connection can be rejected depending on the implementation.

Bit(4,5): LCT Duration

Four durations of Link Confidence Test are allowed and Bit(4) and Bit(5) specify the LCT duration suggested by this port. If the suggested values from the two ports are different, then the longer duration is used for the Link Confidence Test. The LCT duration is encoded into Bit(4) and Bit(5) as shown in Table VI.

TABLE VI. LCT Duration Encoding

LCT Duration	Bit(4)	Bit(5)
Short	0	0
Medium	0	1
Long	1	0
Extended	1	1

The Short LCT duration is used when there is no recent history of excessive link errors. The Medium LCT duration is used when a failure in LCT occurred. The Long LCT duration is used after the rejection of a link due to an excessive Link Error Rate. And finally, the Extended LCT duration is used when LCT is used to withhold an undesirable connection.

Bit(6): MAC Available for LCT

Bit(6) is set to indicate that a MAC will be placed at this end of the connection during the Link Confidence Test.

If the other port on the link does not have a MAC available for the Link Confidence Test, then this end may optionally source valid PDUs.

If this end cannot connect to a MAC during the Link Confidence Test, then this PHY device is configured to repeat any received PDUs.

Note that LCT is performed after Bit(6) is communicated, during the Next state.

Bit(7): LCT Failed

Bit(7) is set to indicate that the Link Confidence Test was failed by this end of the connection.

If either side signals that the LCT failed, the connection is restarted and a long Link Confidence test is used next time.

Bit(8): MAC for Local Loop

Bit(8) is set to indicate that this end of the connection will provide a MAC for the MAC Local Loop. MAC Local Loop may optionally be used to verify MAC recovery processes, token passing and the neighbor notification process.

If neither side set Bit(8), then the MAC Local Loop is omitted. If this end does not support MAC Local Loop and the other end does, this end of the connection has the option of not performing the MAC local Loop.

Bit(9): MAC on Port Output

Bit(9) is set to indicate that this end of the connection intends to place a MAC in the output token path.

5.3 PCM TIMERS AND TIMER EXPIRATION VALUES

There are only a few timers used in PCM to determine the length of certain operations and the time in which an appropriate response is expected However, one timer can have different expiration values depending on the state of the PCM software.

For most of the time, the timers are used to measure the maximum timer to wait for the reception of certain line states, the minimum duration to transmit certain line states, the duration for LCT, etc. Therefore, the timer operations help to ensure the two PCMs are synchronized.

The following timers are employed in PCM:

TPC

Physical Connection Timer is the main timer that PCM uses. It is used to ensure state transitions proceed at the desired rate.

TID

The Timer for Idle Detection is used by PCM to measure the time of continuous ILS reception.

TNE

The Timer for Noise Event is used by PCM to detect the length of noise events. If an excessive number of noise is received, the PCM state machine can optionally reinitialize the connection by moving to the Break state.

The rest of this section lists the timer expiration values used in the PCM state machine. All of the expiration values are used for TPC timer unless specified otherwise. The default value is also highlighted for the ease of referencing.

PC__React: 3.0 ms

PC__React states the maximum timer for PCM to make a state transition to the Break state when the connection needs to be restarted. The reinitialization of the connection is needed upon the reception of Quiet line state, when a fault condition is detected, or PC__Start is presented.

TB__Min: • ms

The minimum time that PCM transmits Quiet symbols and receives Quiet line state during Break state. This value is rather large because PCM has to wait for the other end to response.

TB__Max: 50 ms

The maximum time the state machine is allowed to remain in the Break state before an error flag (BS__Flag) is set to indicate that the PCM is stuck in the Break state.

C__Min: 1.6 ms

The minimum time the state machine is required to send Halt symbols after the reception of Halt line state during the Connect state. This timer is used to assure that the other end of the connection has recognized the Halt symbols being transmitted in the Connect state.

LS__Min: 480 ns

The minimum time of continuous reception of Idle line state before Idle line state is recognized by the PCM state machine. The duration of LS__Min is greater than or equal to 12 symbol times to ensure robustness. This expiration value to be used in the Next state and the Active state. It is measured by the TID timer.

TL__Min: 0.3 ms

TL__Min is the minimum time to transmit a line state before advancing to the next state. It is used in the following states: Next, Signal, Join, Verify and Active. The TL__Min value is set to twice the time required for a line state to be recognized by the PHY device in the other end of the link.

T__Out: 100 ms

T__Out specifies the signaling timeout value. It is defined as the minimum time the state machine is required to remain in a state to wait for the line state reception before reinitialization of the connection.

The expiration of T__Out indicates that a line state change is expected but did not happen, the connection has failed and needs to be re-started in the Break state.

T__Next(n): 100 ms

T__Next(n) is the same as the T__Out values but it is used in the Next state since LCT and MAC Local Loop is to be performed during the Next state. T__Next(n) specifies the timeout value for the nth bit in the signaling sequence. T__Next(7) and T__Next(9) specifies a different timeout value other than the default ones.

T__Next(7): LCT Duration

The T__Next(7) value is the negotiated value of the Link Confidence Test duration after the signaling of Bit(4) and Bit(5). It can take on one of the following values: LC__Short (50 ms), LC__Medium (500 ms), LC__Long (5.0 sec) or LC__Extended (50 sec).

T__Next(9): 200 ms

The maximum time for the optional MAC Local Loop to be performed. This timer is used to prevent deadlock while allowing sufficient time for MAC recovery process completion and exchange of neighbor information frames.

NS__Max: 1.3 ms

The maximum time that noise is tolerated before the connection is considered to be unreliable and needs to be re-started. This timeout value is based on the TNE timer.

5.4 SIGNALS

A signal is used to initiate a state change within PCM. It is also used to communicate with other SMT entities. The signals can be generated by PCM or other entities.

The following signals are used in PCM:

PC__Start

A signal that is set by the Entity Coordination Management Entity (ECM) to PCM. PC__Start is used to signal the PCM to initialize a connection.

PC__Start is also signaled by PCM when the Link Confidence Test fails and the connection is re-initialized.

PC__Maint

A signal that is set by a higher-level management entity to PCM. PC__Maint is used to signal PCM to enter the Maintenance state.

PC__Trace

A signal that is set by ECM to PCM. PC__Trace is used to signal PCM to enter the Trace State.

PC__Stop

A signal that is set by ECM to PCM. PC__Stop is used to signal PCM to enter the Off state.

PC__Enable

A signal that is set by a management entity to PCM. PC__Enable is used to signal PCM to move from the Maintenance state to the Off state.

PC__Disable

A signal that is set by a management entity to PCM. PC__Disable is used to signal PCM to move from any state to the Maintenance state.

PC__Signal

A signal that is set by PCM for its internal use during the Signaling Sequence. PC__Signal is used to indicate that the next value is available and ready for transmission.

PC__PDR

A signal that is set by PCM for its internal use. PC__PDR is used to indicate that a PDR is to be transmitted.

PC__Join

A signal that is set by PCM for its internal use. PC__Join is used to indicate that the Signaling Sequence has been completed successfully and the Join Sequence is started.

5.5 PCM FLAGS AND VARIABLES

A flag is a variable that shall take one of two values: set(1) or cleared(0). Flags are used to reflect the status of the state machine and also to signal other entities of the PCM status.

Variables can take on a wider but still a limited set of values. Variables serve the same function of Flags.

The following is a list of flags and variables used in PCM:

PC__MAC__LCT

A flag that is available internally to PCM. This flag is used to indicate that a MAC will be used for the Link Confidence test.

PC__MAC__Loop

A flag that is available internally to PCM. This flag is used to indicate that the MAC Local Loop will be performed before the connection is made active.

CF__MAC

A flag that is set by the Configuration Control Management (CCM) to PCM. This flag is used to indicate that a MAC is available for the Link Confidence test or MAC Local Loop.

BS__Flag

The Break State Flag (BS__Flag) is set by PCM to indicate that the PCM state machine is not leaving the Break state in an expected time interval and a problem is suspected. It can be used by other management agencies to indicate a problem in PCM or the link that needs to be resolved.

LEM__Fail

A flag that is set by PCM to other management entities. It is set to indicate that the Link Error Rate exceeds the LER__Cutoff threshold. The flag is cleared when the Link Error Rate Threshold test is passed. It is used to remove connection with excessive Link Error Rate.

PC__Type

A variable that is set by the higher level management agency to PCM. It is used to specify the type of port being managed by the PCM.

Four different ports are defined:

A The port in a Dual Attachment Station or Concentrator which attaches to the Primary In and Secondary Out when attaching to the dual ring.

B The port in a Dual Attachment Station or Concentrator which attaches to the Secondary In and Primary Out when attaching to the dual ring.

S The port in a Single Attachment Station (SAS) or one of the port in a Single Attachment Concentrator (SAC).

M The port in a Concentrator that is used to connect with a SAS or SAC.

PC__Neighbor

A variable that is set by PCM to other management entities. This variable is set to indicate the PC__Type of the PHY at the other end of the connection. PC__Neighbor is set during the signaling sequence.

PC__Neighbor can have one of five values: A, B, S, M, or None.

PC__Mode

A variable that is set from PCM to other management entities. This variable is set after the signaling sequence has been completed to indicate the mode of physical connection that has been performed.

PC__Mode can have one of three values:

Peer PC__Mode is set to Peer when neither the port under control or the port at the other end are of type M. It indicates that this connection exists within the trunk ring.

Tree PC__Mode is set to Tree when one of the port is of type M. It indicates that the connection exists with a concentrator tree.

None The connection is neither of the two previous values. PC__Mode is set to None when the connection type is yet unknown.

PC__Withold

A variable that is set by PCM to other management entities. This variable is used to indicate the reason the connection did not get incorporated in the ring.

PC__Withold can have one of the following three values: None, Port M to Port M, or Other incompatible Port Types.

Maint__LS

A variable that is set by other management entities to PCM. This variable is used to indicate the symbol stream to be transmitted when the PCM is in the Maintenance state.

Maint__LS has one of the following values: Quiet, Halt, Idle, Master or PDR.

PC__LS

A variable that is set by PCM to another management entity. This variable is set to indicate the line states received by the PHY.

PC__LS can have one of the following values: QLS, HLS, MLS, ILS, ALS, NLS or LSU.

PC__LCT__Fail

A variable that is set by PCM to other management entities. This variable is used to indicate the number of consecutive failures of the Link Confidence Test.

n

A variable that is set by PCM for its internal use. This variable is used to indicate the number of the next value to be signaled in the Next state and the current value being signaled in the Signal state.

LIFE SUPPORT POLICY

NATIONAL'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT OF NATIONAL SEMICONDUCTOR CORPORATION. As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and whose failure to perform, when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.
2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

National Semiconductor Corporation
2900 Semiconductor Drive
P.O. Box 58090
Santa Clara, CA 95052-8090
Tel: 1(800) 272-9959
TWX: (910) 339-9240

National Semiconductor GmbH
Livry-Gargan-Str. 10
D-82256 Fürstenfeldbruck
Germany
Tel: (81-41) 35-0
Telex: 527649
Fax: (81-41) 35-1

National Semiconductor Japan Ltd.
Sumitomo Chemical
Engineering Center
Bldg. 7F
1-7-1, Nakase, Mihama-Ku
Chiba-City,
Ciba Prefecture 261
Tel: (043) 299-2300
Fax: (043) 299-2500

National Semiconductor Hong Kong Ltd.
13th Floor, Straight Block,
Ocean Centre, 5 Canton Rd.
Tsimshatsui, Kowloon
Hong Kong
Tel: (852) 2737-1600
Fax: (852) 2736-9960

National Semiconductores Do Brazil Ltda.
Rue Deputado Lacorda Franco
120-3A
Sao Paulo-SP
Brazil 05418-000
Tel: (55-11) 212-5066
Telex: 391-1131931 NSBR BR
Fax: (55-11) 212-1181

National Semiconductor (Australia) Pty, Ltd.
Building 16
Business Park Drive
Monash Business Park
Nottinghill, Melbourne
Victoria 3168 Australia
Tel: (3) 558-9999
Fax: (3) 558-9998

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

TI products are not authorized for use in safety-critical applications (such as life support) where a failure of the TI product would reasonably be expected to cause severe personal injury or death, unless officers of the parties have executed an agreement specifically governing such use. Buyers represent that they have all necessary expertise in the safety and regulatory ramifications of their applications, and acknowledge and agree that they are solely responsible for all legal, regulatory and safety-related requirements concerning their products and any use of TI products in such safety-critical applications, notwithstanding any applications-related information or support that may be provided by TI. Further, Buyers must fully indemnify TI and its representatives against any damages arising out of the use of TI products in such safety-critical applications.

TI products are neither designed nor intended for use in military/aerospace applications or environments unless the TI products are specifically designated by TI as military-grade or "enhanced plastic." Only products designated by TI as military-grade meet military specifications. Buyers acknowledge and agree that any such use of TI products which TI has not designated as military-grade is solely at the Buyer's risk, and that they are solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI products are neither designed nor intended for use in automotive applications or environments unless the specific TI products are designated by TI as compliant with ISO/TS 16949 requirements. Buyers acknowledge and agree that, if they use any non-designated products in automotive applications, TI will not be responsible for any failure to meet such requirements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

Products		**Applications**	
Audio	www.ti.com/audio	Communications and Telecom	www.ti.com/communications
Amplifiers	amplifier.ti.com	Computers and Peripherals	www.ti.com/computers
Data Converters	dataconverter.ti.com	Consumer Electronics	www.ti.com/consumer-apps
DLP® Products	www.dlp.com	Energy and Lighting	www.ti.com/energy
DSP	dsp.ti.com	Industrial	www.ti.com/industrial
Clocks and Timers	www.ti.com/clocks	Medical	www.ti.com/medical
Interface	interface.ti.com	Security	www.ti.com/security
Logic	logic.ti.com	Space, Avionics and Defense	www.ti.com/space-avionics-defense
Power Mgmt	power.ti.com	Transportation and Automotive	www.ti.com/automotive
Microcontrollers	microcontroller.ti.com	Video and Imaging	www.ti.com/video
RFID	www.ti-rfid.com		
OMAP Mobile Processors	www.ti.com/omap		
Wireless Connectivity	www.ti.com/wirelessconnectivity		
	TI E2E Community Home Page		e2e.ti.com

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265