INTERNATIONAL STANDARD

ISO/IEC 9314-5

First edition
1995-02-01

Information technology — Fibre Distributed Data Interface (FDDI) —

Part 5:
Hybrid Ring Control (HRC)

Technologies de l'information — Interface de données distribuées sur fibre (FDDI) —

Partie 5: Commande hybride par anneau (HRC)

Reference number
ISO/IEC 9314-5:1995(E)

Contents

International Organization for Standardization
Case postale 56 • CH-1211 Genève 20 • Switzerland
Printed in Switzerland

Annexes

Figures

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through the technical committees established by the respective organization to deal with particular fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

In the field of information technology, ISO and IEC have established a joint technical committee ISO/IEC JTC 1. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75% of the national bodies casting a vote.

International Standard ISO/IEC 9314-5 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology,* Subcommittee SC 25, *Interconnection of information technology equipment.*

ISO/IEC 9314 consists of the following parts, under the general title *Information technology — Fibre Distributed Data Interface (FDDI):*

— *Part 1: Token Ring Physical Layer Protocol (PHY)*

— *Part 2: Token Ring Media Access Control (MAC)*

— *Part 3: Physical Layer, Medium Dependent (PMD)*

— *Part 5: Hybrid Ring Control (HRC)*

Annexes A to F of this part of ISO/IEC 9314 are for information only.

Information technology — Fibre Distributed Data Interface (FDDI) —

Part 5:
Hybrid Ring Control (HRC)

1 Scope

This part of ISO/IEC 9314 specifies a hybrid ring control (HRC) protocol which provides a mode of operation in which both packet switched and isochronous data are transmitted within the same special frame structure, called a cycle. HRC is designed to operate with the existing media access control (MAC), physical layer (PHY), and physical medium dependent (PMD) layers of the FDDI protocol.

The HRC is composed of the hybrid multiplexer (H-MUX) and the isochronous media access control (I-MAC) protocols. The H-MUX integrates packet and isochronous data into cycles which it transmits onto and receives from the medium using the services of the physical layer. The I-MAC provides separate transmission channels for the transfer of user isochronous data streams. The format, clocking and synchronization of cycles, and the operation and interfaces of the H-MUX and I-MAC are defined by this part of ISO/IEC 9314. These interfaces include the interface to the FDDI station management (SMT) protocol.

The HRC is designed to support various transmission rates, from 100 Mbps upwards, in increments of 6,144 Mbps. All transmission rate dependent parameters defined in this part of ISO/IEC 9314 assume a transmission rate of 100 Mbps.

Stations composed of FDDI and HRC entities are referred to as FDDI-II stations. The FDDI packet MAC (P-MAC) and the HRC components, and their architectural relationship to LLC and a circuit switching Multiplexer (CS-MUX) are illustrated in figure 1. This figure does not imply an implementation configuration

FDDI-II networks consist of FDDI-II stations. Interoperability between FDDI and FDDI-II stations on the same network is provided in HRC basic mode, which only supports packet transmission.

The set of FDDI standards, ISO/IEC 9314, specifies the interfaces, functions, and operations necessary to ensure interoperability between conforming FDDI implementations. This part of ISO/IEC 9314 specifies a hybrid ring control protocol: HRC. Conforming implementations may employ any design technique that does not violate interoperability.

2 Normative references

The following standards contain provisions which, through reference in this text, constitute provisions of this part of ISO/IEC 9314. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this part of ISO/IEC 9314 are encouraged to investigate the possibility of applying the most recent editions of the standards listed below. Members of IEC and ISO maintain registers of currently valid International Standards.

ISO 9314-1:1989, *Information processing systems — Fibre Distributed Data Interface (FDDI) — Part 1: Token Ring Physical Layer Protocol (PHY).*

ISO 9314-2:1989, *Information processing systems — Fibre Distributed Data Interface (FDDI) — Part 2: Token Ring Media Access Control (MAC).*

ISO/IEC 9314-3:1990, *Information processing systems — Fibre Distributed Data Interface (FDDI) — Part 3: Physical Layer Medium Dependent (PMD).*

ISO/IEC 9314-7:----[1], *Information technology — Fibre Distributed Data Interface (FDDI) — Part 7: Physical Layer Protocol-2 (PHYS-2).*

ISO/IEC 9314-8:----[1], *Information technology — Fibre Distributed Data Interface (FDDI) — Part 8: Media Access Control-2 (MAC-2).*

ISO 8802-2:1994, *Information technology — Telecommunications and information exchange between systems — Local and metropolitan area networks — Specific requirements — Part 2: Logical link control.*

3 Definitions

For the purpose of this part of ISO/IEC 9314, the following definitions apply:

3.1 basic mode: An FDDI-II network operating in basic mode supports FDDI token ring operation only, that is, only the packet switching service is provided. The data unit transmitted on the medium in basic mode is the FDDI frame.

3.2 channel: The term 'channel' is a synonym for 'transmission channel'.

3.3 circuit: A circuit is a bidirectional communications capability provided over a continuous isochronous channel(s) between two or more CS-MUX level entities.

3.4 circuit switching: Circuit switching is the service that provides and manages a set of circuits.

3.5 circuit switching multiplexer (CS-MUX): A CS-MUX multiplexes and demultiplexes circuits onto transmission channels for transmission.

3.6 connection: A connection is a concatenation of circuits and other functional units set up to provide for the transfer of signals between two or more points in a telecommunications network.

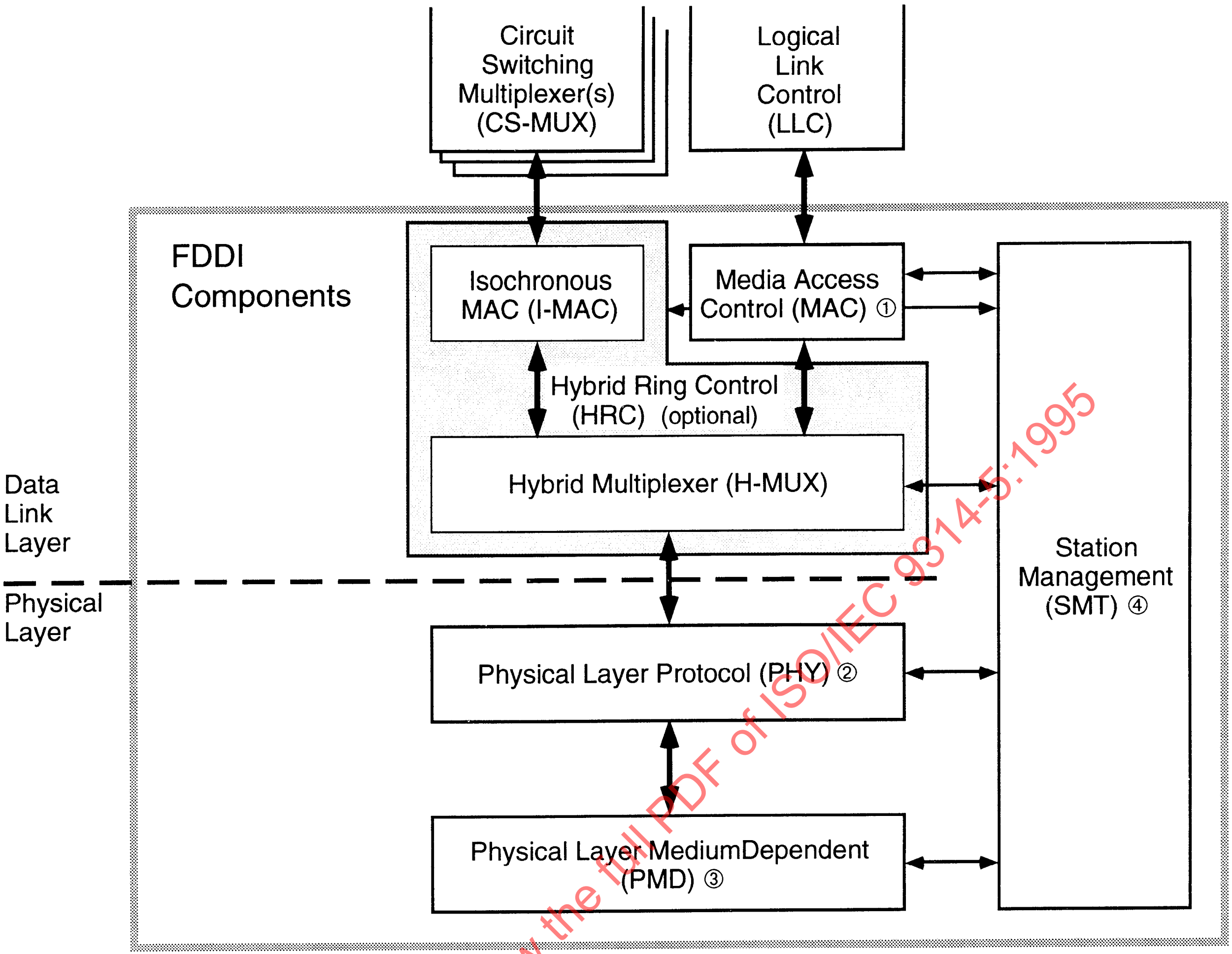
3.7 cycle: The cycle is the HRC frame. It has a duration of 125 µs and nominally carries 3 120 symbols at 100 Mbps.

3.8 cycle control field: The cycle control field is a two symbol field in the cycle header. One symbol is for synchronization control while the other is for sequence control. These are used to indicate whether or not cycle synchronization and sequence, respectively, are being maintained. Each of these fields may only be set by the cycle master.

3.9 cycle header: The cycle header begins with the preamble, which establishes the boundary of the 125 µs cycle. The remainder of the cycle header provides synchronization control, sequence control, a cycle sequence field and the cycle programming template.

3.10 cycle master: One ranked monitor in an FDDI-II ring assumes the role of the cycle master. The ring has only one cycle master at a time. The cycle master is responsible for generating and maintaining the cycle structure and the timing of the ring. The cycle master inserts a latency adjustment buffer to adjust the ring size to be an integer multiple of 125 µs. The cycle master is selected by bidding among ranked monitor stations — the monitor with the highest rank becomes the cycle master.

[1] To be published.



① MAC-2 with HRC; MAC or MAC-2 otherwise.

② PHY-2 with HRC; PHY or PHY-2 otherwise.

③ PMD, SMF_PMD, TP-PMD or LCF-PMD.

④ SMT-2 with HRC; SMT or SMT-2 otherwise.

Figure 1 — Structure of FDDI standards

3.11 cycle sequence: Cycle sequence is a scheme for indicating whether or not the correct order of cycle transmission is being maintained during normal hybrid mode operation. The sequence number of each cycle is indicated in the cycle sequence field of the cycle header. Cycle sequence values 1-63 are used to indicate monitor ranking, and values 64-255 are used for sequencing.

3.12 cycle structure: The cycle structure defines the format of the cycle. The cycle is comprised of the preamble, cycle header, dedicated packet group and cyclic groups.

3.13 cyclic groups: The cycle structure contains 16 wideband channels (WBCs), which are byte interleaved with each other. The interleaving scheme physically organizes the WBCs into 96 cyclic groups per cycle, at 100 Mbps. Each cyclic group contains one byte from each WBC. The bytes from each WBC occur in the same position in each cyclic group.

3.14 dedicated packet group (DPG): The dedicated packet group is the part of the cycle structure which provides a minimum packet channel bandwidth of 0,768 Mbps (at 100 Mbps).

3.15 entity: An entity is an active functional agent within an (OSI) layer or sublayer, including both operational and management functions.

3.16 FDDI-II: FDDI-II is a term used to describe a network consisting of stations composed of FDDI and HRC entiities.

3.17 fibreoptics: The technology whereby optical signals from light-generating transmitters are propagated through optical fibre waveguides to light-detecting receivers.

3.18 hybrid isochronous-MAC service access point (HI-SAP): HI-SAPs are the isochronous access points of the H-MUX. They are used by the I-MAC to access the WBCs.

3.19 hybrid mode: An FDDI-II network operating in hybrid mode imposes a cycle structure with a length of 125 µs. The cycle supports a variable rate packet switching service using the FDDI token ring protocol plus a time-division multiplexed circuit switching service. The bandwidth is partitioned as a dedicated packet data channel plus 16 wideband channels which are dynamically allocated for packet data or isochronous use.

3.20 hybrid multiplexer (H-MUX): The hybrid multiplexer is the component that directs the data flow between the packet and isochronous media access control sublayer and the physical Layer.

3.21 hybrid packet-MAC service access point (HP-SAP): The HP-SAP is the access point of the H-MUX. It is used by the P-MAC to access the packet data channel. One HP-SAP exists and makes available a variable data rate between 0,768 Mbps and 99,072 Mbps, dependent on the partitioning of WBCs between circuit switching and packet switching.

3.22 hybrid ring control: Hybrid ring control is the protocol that exists in an FDDI-II station to provide integrated packet and isochronous switching. It consists of the hybrid multiplexer and the isochronous media access control.

3.23 hybrid service access point (H-SAP): H-SAPs are the service access points of the H-MUX. They are divided into one HP-SAP and up to 16 HI-SAPs.

3.24 isochronous: The term ‘isochronous’ indicates the essential characteristic of a time-scale or a signal such that the time intervals between consecutive significant instants either have the same duration or durations that are integral multiples of the shortest duration.

3.25 isochronous MAC service access point (I-SAP): I-SAPs are the access points of the I-MAC. They are used by the CS-MUX to access isochronous transmission channels. One and only one isochronous MAC service access point is associated with a transmission channel. Each I-SAP may be open or closed by SMT.

3.26 isochronous media access control (I-MAC): The I-MAC is the data link sublayer entity which provides isochronous data access to a shared medium local area network.

3.27 LAPD: LAPD is the link access procedure for the ISDN D-channel as defined by CCITT.

3.28 latency adjustment buffer (LAB): The LAB is a component that is required at the cycle master to insure that the ring isochronous channel latency is a multiple of 125 µs.

3.29 media access control (MAC): The MAC is the data link sublayer responsible for scheduling and routeing data transmissions on a shared medium local area network (e.g. an FDDI ring).

3.30 mode: There are two modes of operation in FDDI: basic and hybrid. Basic mode is the mode of operation based on the FDDI frame structure where only Packet switching is supported. Hybrid mode is the mode of operation based on the 125 µs cycle where both packet and circuit switching are supported.

3.31 monitor contention procedure: This is one method by which one monitor station is selected to become the cycle master. Only monitor stations that have a rank may bid, and the station with the highest rank wins the right to become cycle master.

3.32 monitor rank: Rank is a priority scheme for the selection of a cycle master from monitor stations. Rank values range from 0 to 63, with 63 being the highest. monitor stations with a rank of 0 do not participate in the monitor contention procedure. A monitor indicates its rank by loading it into the cycle sequence field of the cycle header during the monitor contention procedure.

3.33 monitor station: Monitor stations in an FDDI-II network are those stations capable of becoming the cycle master. A monitor station is defined as a station containing a hybrid ring control entity with the Class parameter set to MONITOR. All monitor stations contain an LAB and a cycle generate block. A monitor station is permitted to participate in the monitor contention procedure, however it is only permitted to become the cycle master if it has a rank that is greater than zero.

3.34 multi-point circuit: A multi-point circuit is a circuit which exists between more than two CS-MUX entities for multi-point communications.

3.35 non-monitor station: A non-monitor station is a station which does not satisfy the criteria for a monitor station. In general a non-monitor station does not have the capability of becoming the cycle master.

3.36 octet: An octet is a data unit composed of eight ordered bits and is transmitted in FDDI-II as a pair of data symbols.

3.37 packet media access control (P-MAC): The data link sublayer entity responsible for scheduling and routeing data transmissions via a packet data channel on a shared medium local area network.

3.38 packet switching: The service that transfers packetized data. It is provided in FDDI and FDDI-II by the FDDI packet media access control entity (P-MAC).

3.39 physical layer protocol (PHY): The physical layer protocol is responsible for transmitting and receiving a symbol stream on the physical transmission medium.

3.40 protocol data unit (PDU): The unit of data transfer between communicating peer layer entities. It can contain control information, address information, and/or data (e.g., an SDU from a higher layer entity). The FDDI MAC PDUs are tokens and frames.

3.41 programming template: The programming template is a part of the cycle header. The programming template is used by the hybrid multiplexer in each station to determine whether each of the wideband channels is currently assigned to packet or isochronous traffic. The programming template is read by all FDDI-II stations, but its contents may only be modified by the cycle master.

3.42 Q.931: Q.931 is the call control signalling protocol for use on the ISDN D-channel as defined by CCITT.

3.43 receive: This is the action of a station which accepts a token, frame or other symbol sequence from the incoming medium.

3.44 ring: A closed loop consisting of one or more stations connected by a physical medium wherein information is passed sequentially between active stations, each station in turn examining or copying and repeating the information, finally returning it to the originating station.

3.45 services: A set of functions provided by one OSI layer or sublayer entity, for use by a higher layer or sublayer entity or by management entities. Data services are provided to a higher layer or sublayer entity; management services are provided to a management entity.

3.46 service primitive: A service primitive is an access procedure for a service. It is presented at the service interface.

3.47 service data unit (SDU): The unit of data passed between a service user and a service provider by a service primitive. The MAC SDU is the data content of a frame. The PHY SDU is a symbol.

3.48 slave station: A slave station in an FDDI-II network is any station which is not the cycle master. A slave station can be either a monitor station or a non-monitor station.

3.49 station: An addressable logical and physical node in an FDDI network, capable of transmitting, repeating and receiving information. An FDDI station has one or more PHY and PMD entities, zero or more HRC entities, one or more MAC entities, and one SMT entity.

3.50 station management (SMT): The supervisory entity within an FDDI node that monitors and controls the other FDDI entities in the node.

3.51 symbol: The smallest signalling element used by the data link layer (DLL). The symbol set consists of 16 data symbols and 8 control symbols.

3.52 synchronous: A class of data transmission service whereby each requester is preallocated a maximum bandwidth and guaranteed a maximum access time.

3.53 token: An explicit indication of the right to transmit on a shared medium. On a token ring, the token circulates sequentially through the stations in the ring. At any time, it may be held by zero or one station. MAC uses two classes of tokens: restricted and nonrestricted.

3.54 transmission channel: A transmission channel is a portion of a wideband channel. It is used to transfer isochronous data. Transmission channels may be of different sizes. The ISDN basic rate channel, of 64 kbps, is one of the transmission channels supported by this part of ISO/IEC 9314.

3.55 transmit: This is the action of a station which generates a token, frame, Cycle, or other symbol sequence and places it on the outgoing medium.

3.56 wideband channel (WBC): The WBC is a single unit of bandwidth in the HRC that is assigned to either isochronous or packet data use. WBCs may be subdivided into transmission channels.

4 Conventions and abbreviations

4.1 Conventions

The terms SMT, PHY, HRC, I-MAC, P-MAC, CS-MUX, and LLC, when used without modifiers, refer specifically to the local entities. The term LLC unless otherwise qualified refers to any local user of MAC data services, other than SMT. This includes ISO 8802-2.

Low lines (e.g. requested_service_class) are used as a convenience to mark the name of signals, functions, etc., that might otherwise be misinterpreted as independent individual words if they were to appear in text.

The use of period (e.g. MA_UNITDATA.request) is equivalent to the use of low lines except that a period is used as an aid to distinguish modifier words appended to an antecedent expression.

The following symbols are used in state diagrams and associated notes with the meanings indicated

¬ logical not function
| logical or function
& logical and function.

4.1.1 Timing values and timers

For the purpose of the description contained in this HRC standard, all timers are assumed to be initialized with the unsigned twos complement of the target time, in either octets or cycles as applicable. Timers are further assumed to count upward if enabled, expiring when an overflow occurs. All timer comparisons are expressed on the basis of elapsed time. Timers run normally in all states if enabled.

4.2 Abbreviations

CACQ	cycle acquisition
CDG	cyclic data group
CGEN	cycle generation
CS-MUX	circuit switch multiplexer
CXC	cycle exchange
DPG	dedicated packet group
FDDI	Fibre Distributed Data Interface
H-MUX	hybrid multiplexer
H-SAP	hybrid service access point
HI_IND	HI_UNITDATA.indication interface data unit
HI_INV	HI_INVALID.indication
HI_REQ	HI_UNITDATA.request interface data unit
HI-SAP	H-MUX Isochronous service access point
HP_IND	HP_UNITDATA.indication interface data unit
HP_INV	HP_INVALID.indication
HP_REQ	HP_UNITDATA.request interface data unit
HP-SAP	H-MUX packet service access point
HRC	hybrid ring control
I_IND	IM_UNITDATA.indication interface data unit
I-MAC	isochronous media access control
I_REQ	IM_UNITDATA.request interface data unit
I-SAP	isochronous MAC service access point
LAB	latency adjustment buffer
LLC	logical link control sublayer
IMC	isochronous maintenance channel
PDC	packet data channel
PDU	protocol data unit
P-MAC	packet media access control
PH_IND	PH_UNITDATA.indication interface data unit
PH_INV	PH_INVALID.indication
PH_REQ	PH_UNITDATA.request interface data unit
PHY	physical layer
SDU	service data unit
SMT	station management
TNS	new sequence timer
TVS	valid sequence timer
WBC	wideband channel

5 General description

HRC is a protocol which integrates isochronous and packet data on the same FDDI medium. It is designed to operate at 6,144 Mbps increments above and below the base rate of 100 Mbps. This specification describes operation at 100 Mbps.

The HRC protocol is based on special HRC frames called cycles that carry packet and isochronous data. A new cycle, consisting of control and data octets, is generated by the cycle master every 125 μs. The 125 μs cycle was chosen to facilitate interconnection with public switched networks. This cycle is partitioned into four parts: the preamble, the cycle header, the dedicated packet group, and the wideband channel cyclic groups, its structure is illustrated in figure 2.

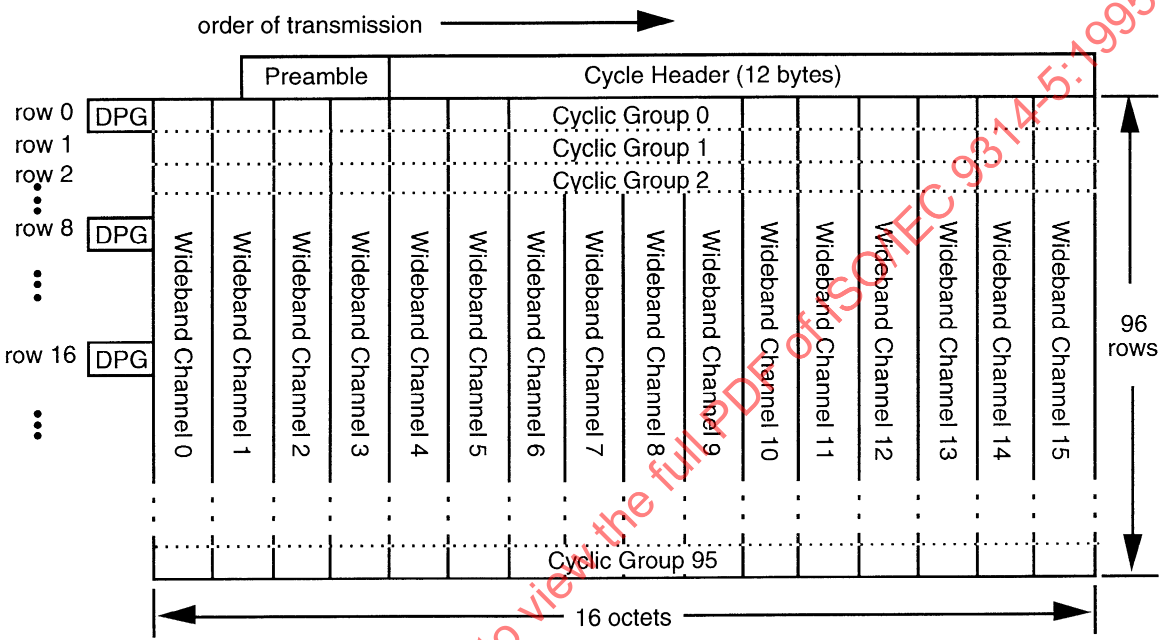


Figure 2 — HRC cycle structure

The data octets of the cycle are divided into a dedicated packet group (DPG) and 16 wideband channels (WBCs). The DPG dedicates 0,768 Mbps of the bandwidth to the FDDI P-MAC. The FDDI MAC is referred to generically in this specification as the packet MAC.

Each WBC provides 6,144 Mbps of bandwidth, and consists of 96 octets interleaved across the cycle. Each WBC can be dynamically allocated to either isochronous or packet data use. When assigned to packet data use, WBCs are combined with the DPG to create the Packet Data Channel. Isochronous WBCs can be subdivided into individual transmission channels. Each data octet of a transmission channel provides 64 kbps of isochronous bandwidth.

The 'dual ring of trees' network topology, specified for use with FDDI, is applicable to FDDI-II. Similarly, the other major parameters defined for FDDI, including the media type, total media path length, bit rate and number of stations supported on a network, also apply to FDDI-II.

There are two modes of operation in an FDDI-II station: basic and hybrid. Basic mode is operation of the ring as an FDDI token ring. This document concentrates on the hybrid mode of operation.

The hybrid mode is the FDDI-II mode of operation where both FDDI token ring operation and isochronous data transfer are integrated onto the same medium. The hybrid mode of operation requires the existence of a hybrid ring control (HRC) entity between the FDDI MAC and the FDDI PHY, as shown in figure 1.

The HRC integrates packet and isochronous data into FDDI-II Cycles, which are transmitted onto the ring isochronously every 125 µs. Based on the cycle header, the hybrid multiplexer (H-MUX) component of the HRC interprets each WBC as either a portion of the packet data channel or as an isochronous channel of 6,144 Mbps.

The packet data channel (PDC) has a minimum rate of 0,768 Mbps, which is provided by the DPG. The PDC can be extended to higher rates, in increments of 6,144 Mbps, by allocating WBCs to it. Its maximum rate of 99,072 Mbps is achieved by allocating all WBCs to the Packet Data Channel.

Each WBC allocated for isochronous use can be subdivided into lower speed transmission channels by the isochronous media access control (I-MAC) component of the HRC. FDDI-II has been designed to allow separate allocator stations for isochronous WBCs, with different allocation policies permitted to coexist on the same ring.

The hybrid mode of operation is initiated and controlled by the cycle master. The cycle master is responsible for generating cycles onto the ring. Stations capable of assuming the cycle master role are known as monitor stations. All other stations on the hybrid ring are referred to as non-monitor stations. The cycle master is chosen from a set of ranked monitor stations before hybrid mode initialization, or during recovery after the 125 µs cycle synchronization has been lost or after out of sequence cycles have been detected. monitor station rankings are used to dynamically select a new cycle master during the monitor contention procedure.

5.1 Traffic types

There are two types of traffic on an FDDI-II network in hybrid mode: Circuit-Switched (isochronous) and Packet-Switched. These types are depicted in figure 3. Circuit-Switched traffic is switched between the H-MUX and the I-MAC. Packet-Switched traffic is switched between the H-MUX and the FDDI MAC, referred to here as the packet MAC (P-MAC) for generality. Packet-Switched traffic medium access is governed by the P-MAC, which permits two types of access: synchronous and asynchronous. Synchronous access uses a pre-assigned portion of the packet data bandwidth. Asynchronous access uses the remaining packet data bandwidth on the reception of an early token. There are two types of tokens: restricted and non-restricted. Packet data access is specified in ISO/IEC 9314-2.

Circuit-Switched	**Packet-Switched**	
Isochronous	Synchronous	Asynchronous:
		Restricted mode Non-Restricted mode

Figure 3 — Hybrid mode traffic types

5.2 Transmission facilities

5.2.1 Transmission channel

WBCs may be subdivided into separate transmission channels to permit simultaneous, independent isochronous dialogues to occur between I-MAC SAPs within the same WBC. FDDI-II transmission channels may be of different data rates, but are never greater than the 6,144 Mbps rate of the WBC.

5.2.2 Channel properties

The properties of an FDDI-II transmission channel are summarized as follows.

a) A channel exists between I-MAC service access points (I-SAPs).

b) A channel is a part or all of a WBC.

c) A channel is contiguous within a WBC.

d) A channel has position within a WBC which is the subject of boundary constraints.

e) A channel has one of a set of sizes.

f) There are two types of coherence class applicable to a channel.

g) A channel has a security label.

h) A channel has an owner.

i) A channel can be de-allocated based upon the allocator's reclamation criteria.

5.2.3 Channel sizes

Channel allocations are permitted in multiples of 64 kbps up to a maximum of 6,144 Mbps (one WBC). The channel is specified by its position within the WBC, the WBC number and the size of the channel and is byte aligned within the WBC. 8 kbps channels are also supported on bit boundaries, 16 kbps channels on even bit boundaries, and 32 kbps channels on quartet (nibble) boundaries. The following list gives examples of channel sizes that are supported by the FDDI-II allocation scheme.

Bits / Cycle	Channel rate (kbps)
1	8
2	16
4	32
8	64
48	384
192	1 536
1+192	1 544
240	1 920
256	2 048
768	6 144

5.2.4 Channel coherence classes

Channel coherence refers to the preservation of sequence integrity for data carried in a channel. Coherence is always maintained within a channel. Coherence is optionally maintained across channels. There are two channel coherence classes:

a) Class 1: coherence is not required across channels.

b) Class 2: coherence is required across channels relative to the cycle.

5.2.5 Circuit

A circuit is defined as the use of a transmission channel between two or more CS-MUX level entities for bi-directional communications. Circuits exist between CS-MUX SAPs.

A multi-point circuit is defined as the use of a transmission channel among more than two CS-MUX level entities for multi-point communications.

5.2.6 Connection

A connection is a concatenation of circuits, circuit switches and other functional units to transfer symbols between two or more end points in a telecommunications network.

5.3 Bandwidth management

Bandwidth management is an SMT function. The description in this subclause is merely informative. A hierarchy of procedures for Bandwidth Management is envisioned,as shown in figure 4. These procedures include WBC management and transmission channel management.

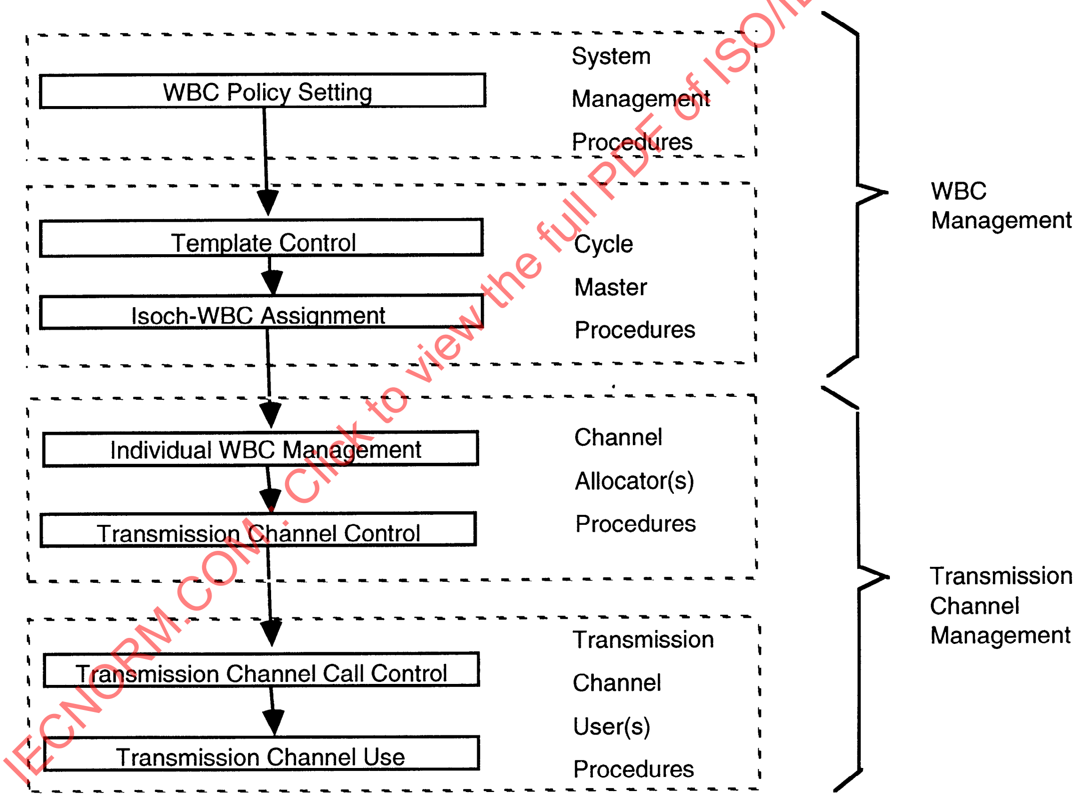


Figure 4 — Bandwidth management hierarchy

5.3.1 Wideband channel management

The procedures for wideband channel Management include the System Management procedure and cycle master procedures.

5.3.1.1 System management procedure

WBC Policy Setting is the System Management procedure used for defining and conveying to the FDDI-II cycle master SMT the maximum number of WBCs that may be allocated for isochronous use. This sets the upper limit on the maximum number of isochronous WBCs that can be administered by a particular individual WBC management entity.

5.3.1.2 Cycle master procedures

There are two cycle master procedures concerned with WBC Management:

a) Template Control: This is the cycle master SMT procedure that instructs the H-MUX to change the cycle header programming template, based on the WBC policy in force. See 5.4 for a discussion of the cycle master and the programming template.

b) Isochronous WBC Assignment: This is the FDDI-II management procedure that assigns or releases a WBC to or from a specific WBC management entity (called a Channel Allocator). The assignment is made to the Channel Allocator for its use and control. This procedure requires interaction between a Channel Allocator SMT and the cycle master SMT.

5.3.2 Transmission channel management

The procedures for transmission channel management include Channel Allocator procedures and transmission channel User procedures.

5.3.2.1 Channel allocator procedures

There are two Channel Allocator procedures:

a) Individual WBC management: This is the procedure used to control the transmission channel bandwidth rates that are available to requesting user stations.

b) Transmission channel control: This is the procedure used for requesting, releasing and supervising the use of a transmission channel.

5.3.2.2 Transmission channel user procedures

a) Transmission channel call control: This is the procedure used for establishing, releasing and supervising a call on an allocated transmission channel.

b) Transmission channel use: This is the procedure used for transferring isochronous information across a transmission channel connection.

5.4 Station structure

An FDDI-II ring has two types of stations: monitor and non-monitor. Monitor stations are those stations capable of becoming the cycle master, and are the only stations permitted to generate cycles. There may be more than one monitor station in a ring, however only one monitor may be the cycle master at any given time. The monitor station with the highest rank becomes the cycle master during the monitor contention procedure. The cycle master is required to establish and maintain the hybrid mode of operation. Non-monitor stations are all other stations that are not capable of becoming the cycle master.

5.4.1 Monitor station

A monitor station, illustrated in figure 5, incorporates the Physical layer protocol (PHY), Packet media access control (P-MAC) and Logical Link Control (LLC) required by FDDI. Additionally, the hybrid multiplexer (H-MUX), isochronous media access control (I-MAC), and latency adjustment buffer (LAB) are required by monitor stations. The LAB is operational in the cycle master and may optionally be operational in the other monitor stations.

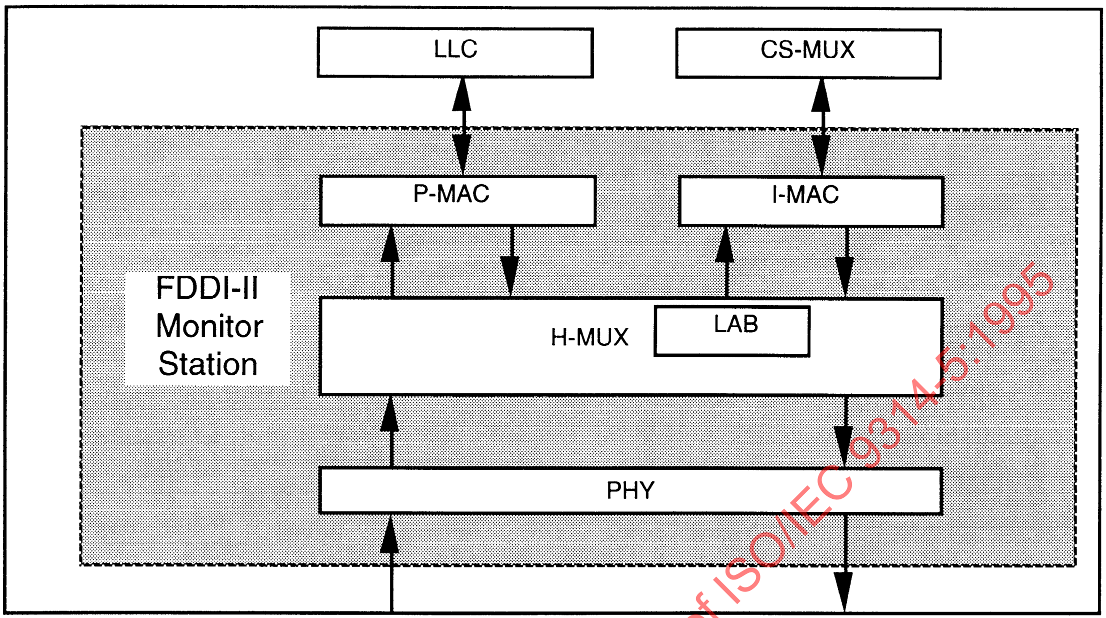


Figure 5 — Data flow through an FDDI-II monitor station

One monitor station becomes the cycle master for hybrid mode initialization or during the hybrid mode monitor contention procedure. While in basic mode, two methods are available for selection of the cycle master. In the first, the system management can directly instruct a monitor station to initialize the ring in hybrid mode. In the second, a monitor station can set a timer and wait for the ring to switch to hybrid mode. In this case, if the timer expires, the monitor station begins the contention process to select the cycle master.

The cycle master is responsible for generating the programming template at the beginning of each cycle and inserting the LAB into the ring to insure that the delay around the ring is an integer multiple of 125 µs. The programming template carries information which associates each WBC with either the packet data channel or with isochronous applications.

NOTE 1 The latency adjustment buffer is inserted at the cycle master station to insure that the delay around the ring for Isochronous data is an integer multiple of 125 µs. It is desirable however, in certain applications, for the H-MUX to present isochronous WBCs to the I-MAC before inserting the data in the LAB. In any case, data at the cycle master for transmission onto the medium is inserted after the LAB.

5.4.2 Non-monitor station

A non-monitor station, illustrated in figure 6, also incorporates the PHY, P-MAC and LLC required by FDDI, and the HRC H-MUX and I-MAC. Non-monitor stations, however, do not have the LAB, nor do they have the capability to initiate cycles.

5.4.3 H-MUX

The H-MUX is the HRC entity responsible for the management of transitions between basic and hybrid modes, and for the control and maintenance of hybrid mode cycle synchronization. The H-MUX also controls the flow of data between the P-MAC, I-MAC and PHY entities in both modes. In basic mode data is passed directly between the P-MAC and the physical layer. In hybrid mode, the H-MUX multiplexes and demultiplexes the I-MAC's isochronous data and the P-MAC's packet data onto or out of the physical layer. The functional architecture of H-MUX is illustrated in figure 7. The shaded blocks (CGEN and LAB) are only required in a monitor station.

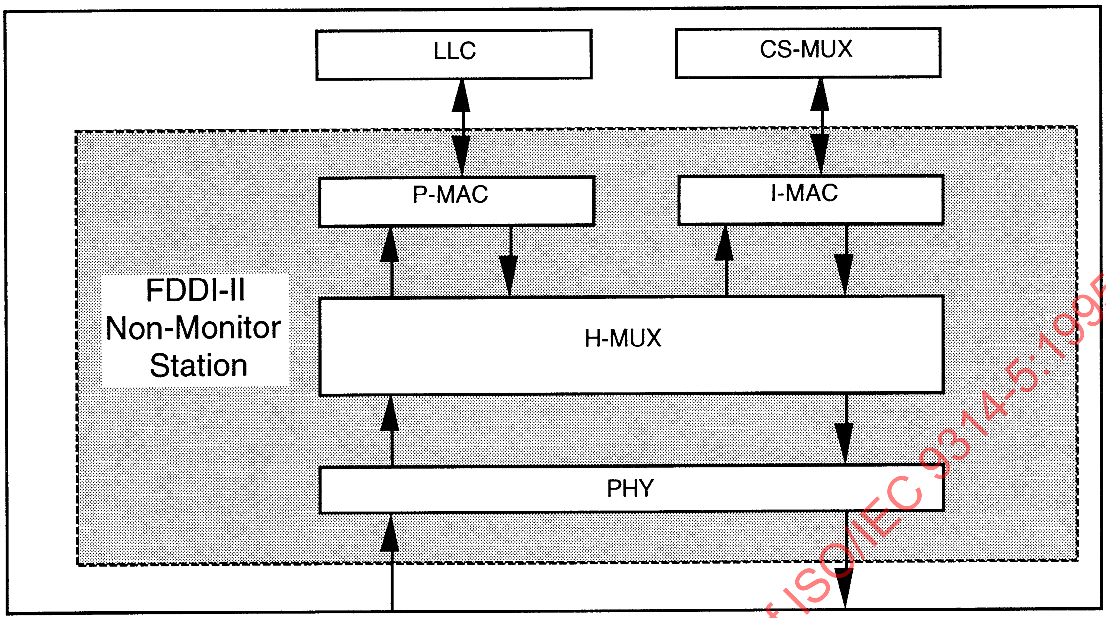


Figure 6 — Data flow through an FDDI-II non-monitor station

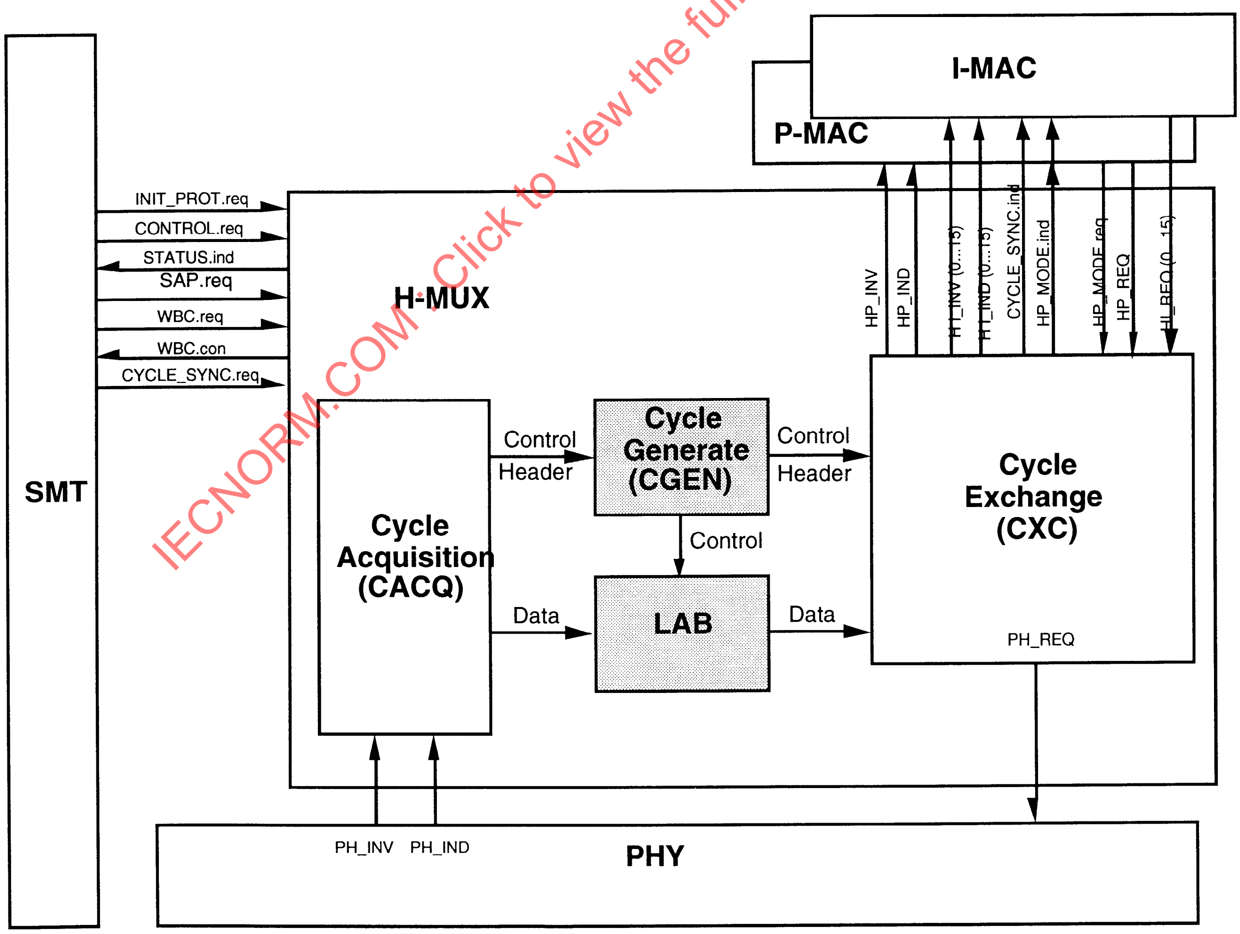


Figure 7 — Architectural block diagram of the H-MUX

The H-MUX in a non-monitor station includes only the cycle acquisition and cycle exchange functional blocks. Clause 8 describes their internal composition.

The H-MUX indicates the current mode of operation to the local MAC entities via the CYCLE_MODE.ind primitive. During hybrid mode operation H-MUX sends the CYCLE_SYNC.ind signal for each cycle received so as to announce the beginning of the cyclic groups to the I-MAC. The I-MAC uses the CYCLE_SYNC.ind signal to locate WBCs for reading and writing data symbol pairs. The Cycle-Sequence value is also passed to the I-MAC as a parameter of CYCLE_SYNC.ind signal.

The programming template, from each received cycle, is copied by the cycle acquisition block for use as input to the WBC template filter state machines. The programming template identifies the contents of each WBC to be either packet or isochronous data.

Hybrid service access points (H-SAPs) are the H-MUX access points for use by the I-MAC and P-MAC. There is one HI-SAP per Isochronous WBC for the I-MAC and a single HP-SAP for the P-MAC.

Each packet data symbol received from the P-MAC in an HP_REQ service primitive is inserted by cycle exchange into a position within the current cycle which corresponds to either a packet WBC or the dedicated packet group (DPG). Similarly each symbol received from either a packet WBC or the DPG is delivered by cycle exchange to the P-MAC using the HP_IND service primitive.

Each isochronous data symbol received from an I-MAC in an HI_REQ service primitive is inserted by cycle exchange into a position within the current cycle which corresponds to the appropriate Isochronous WBC. Similarly each data symbol received from an Isochronous WBC is delivered to the I-MAC using the HI_IND service primitive, providing the HI-SAP corresponding to that WBC is open. If an HI-SAP is closed the Isochronous data for that WBC is repeated.

See clause 6 for further information on Interface Services.

5.4.3.2 Monitor station H-MUX

In order to provide cycle master capability, the H-MUX in a monitor station shall include the cycle generation (CGEN) and latency adjustment buffer (LAB) functions in addition to the functional blocks required by a non-monitor station. Clause 8 describes the internal composition of these blocks.

The LAB provides the buffering required by a cycle master to ensure that the ring contains an integral number of cycles. Received data is written into the LAB by the cycle acquisition process and read out by the cycle exchange process.

The cycle generator provides the extra control functionality required by a monitor for it to have the capability to source cycles. The tasks which need to be performed by the cycle generator therefore include: deciding when to source and when to repeat cycles, controlling the LAB, providing the facility for SMT to assign WBCs, and synchronization of transmit 8 kHz timing to an external reference.

In a cycle master, SMT maintains the programming template and provides it to the cycle master as necessary. The cycle master transmits the programming template in the Cycle-Header with a cycle sequence number. The cycle master reads returning cycle header fields and informs its local SMT entity of ring error conditions. The cycle master also operates in the same way as other stations for data transfer functions.

5.4.4 I-MAC

The I-MAC is responsible for routeing the isochronous channels in use at this station to and from the CS-MUX. Isochronous channels (within open HI-SAPs) not in use at this station are looped back by I-MAC to H-MUX, so that they may be repeated. Figure 8 illustrates the architecture of the I-MAC.

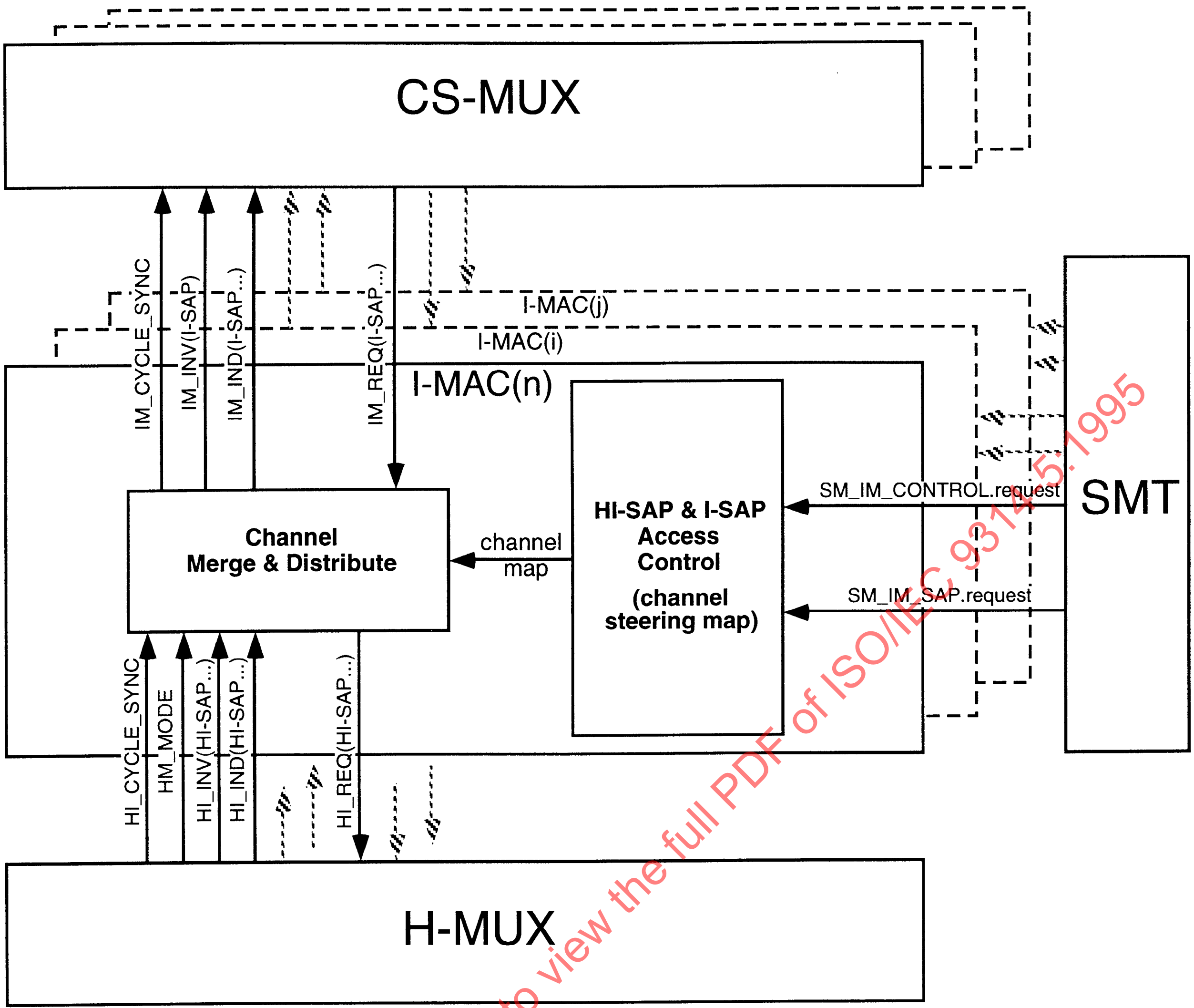


Figure 8 — Architectural block diagram of the I-MAC

The Isochronous service access point (I-SAP) is the I-MAC access point for the CS-MUX. There is one I-SAP per channel allocated at this I-MAC.

The steering map provided by SMT to I-MAC identifies the current open channels (open I-SAPs) within the open isochronous WBCs (open HI-SAPs). The I-MAC receives all symbols in an open HI-SAP from the H-MUX, and separates the channels based on the steering map. Using the cycle sequence signal, the I-MAC delivers these channels to the CS-MUX via the I_IND service primitive . In the opposite direction, I-MAC receives data for the channels from the CS-MUX with I_REQ service primitives.

6 HRC services

This clause describes the services of the HRC. The following subclauses specify the services as follows:

- 6.1 PHY to H-MUX Services
- 6.2 H-MUX to MAC Services
- 6.3 H-MUX to P-MAC Services
- 6.4 H-MUX to I-MAC Services
- 6.5 H-MUX to SMT Services

6.6 I-MAC to CS-MUX Services

6.7 I-MAC to SMT Services

6.1 PHY to H-MUX services

This subclause specifies the services provided at the interface between the PHY and the H-MUX entities, to support the exchange of PDUs among peer H-MUX entities. Additional detail is provided in ISO/IEC 9314-1 concerning conditions that generate these primitives and PHY actions upon receipt of H-MUX-generated primitives.

The following primitives are defined:

PH_UNITDATA.request
PH_UNITDATA.indication
PH_INVALID.indication

All primitives defined in this subclause are mandatory.

The description of each primitive includes a description of the information that shall be passed between the PHY and H-MUX entities.

These services shall be 'synchronous', so that each PH_UNITDATA.indication causes exactly one PH_UNITDATA.request. Depending upon the current internal configuration of the node, the PH_UNITDATA.request may be returned to the same PHY or to a different PHY.

6.1.1 PH_UNITDATA.request

This primitive transfers the symbol data stream to PHY from H-MUX.

6.1.1.1 Semantics of the primitive

```
PH_UNITDATA.request          (
                             PH_Request(symbol)
                             )
```

The symbol specified by PH_Request (symbol) shall be one of the following:

H, I, J, K, L, n, R, S, T and optionally Q or V, from the set of symbols defined in the FDDI standard on PHY.

PH_Request(Q or V) is not required in implementations where the Repeat Filter function is located prior to the PH_UNITDATA.indication interface.

6.1.1.2 When generated

H-MUX shall generate one PH_UNITDATA.request for each PH_UNITDATA.indication received from PHY.

6.1.1.3 Effect of receipt

The effect of receipt of this primitive by PHY is not specified.

6.1.2 PH_UNITDATA.indication

This primitive transfers the symbol data stream from PHY to H-MUX.

6.1.2.1 Semantics of the primitive

PH_UNITDATA.indication (
PH_Indication(symbol)
)

The symbol specified by PH_Indication (symbol) shall be one of the following:

H, I, J, K, L, n, R, S, T and optionally Q or V, from the set of symbols defined in the FDDI standard on PHY.

PH_Indication(Q or V) is not required in implementations where the Repeat Filter function is located prior to the PH_UNITDATA.indication interface.

6.1.2.2 When generated

PHY shall generate a PH_UNITDATA.indication for each decoded symbol received from PMD. This indication shall be generated once every symbol period.

6.1.2.3 Effect of receipt

Upon receipt of this primitive, H-MUX shall accept a symbol from PHY, process it, and generate a corresponding PH_UNITDATA.request to PHY, conveying the resulting output symbol.

6.1.3 PH_INVALID.indication

This primitive indicates to H-MUX that continuity of the received symbol stream has been compromised.

6.1.3.1 Semantics of the primitive

PH_INVALID.indication (
PH_Invalid
)

The PH_Invalid parameter shall indicate that the PH_UNITDATA.indication symbol stream is invalid.

6.1.3.2 When generated

PHY shall generate this primitive whenever it detects that continuity of the received symbol stream has been compromised.

6.1.3.3 Effect of receipt

Receipt of this primitive by H-MUX while it is in basic mode shall cause the H-MUX to pass an HP_INVALID primitive to the P-MAC.

Receipt of this primitive by H-MUX while it is in hybrid mode shall cause the H-MUX to output HP_INVALID and HI_INVALID primitives to the P-MAC and I-MAC. The Receive function of H-MUX shall return to basic mode until another cycle is received.

6.2 H-MUX to MAC services

This subclause specifies the services provided at the interface between the H-MUX and both the I-MAC and P-MAC entities to allow H-MUX to convey its mode of operation to the local MAC entities.

The following primitive is defined:

HM_MODE.indication

6.2.1 HM_MODE.indication

The function of this primitive is to provide information on the current mode of operation of H-MUX to the local MAC entities.

6.2.1.1 Semantics of the primitive

HM_MODE.indication (
HM_mode
)

The HM_mode parameter shall set to 'basic' during basic mode operation, 'hybrid' during hybrid mode operation whenever the H-MUX cycle control state machine is in the SLAVE or STANDBY states, and 'master' during hybrid mode operation whenever the H-MUX cycle control state machine is in the RESYNCH or MASTER states.

6.2.1.2 When generated

H-MUX continuously sends an indication of its mode to the local MAC entities.

6.2.1.3 Effect of receipt

The effect of receipt of this primitive by the local MAC entities is unspecified.

6.3 H-MUX to P-MAC services

This subclause specifies the services provided at the interface between the P-MAC and H-MUX entities to allow P-MAC to exchange P-MAC protocol data units with peer P-MAC entities. Additional detail is provided in the FDDI standard on MAC concerning conditions that generate these primitives and P-MAC actions upon receipt of H-MUX generated primitives.

The following primitives are defined:

HP_UNITDATA.request
HP_UNITDATA.indication
HP_INVALID.indication
HP_MODE.request

All primitives defined in this subclause are mandatory.

The description of each primitive includes a description of the information that is passed between the P-MAC and H-MUX entities.

These services shall be 'synchronous', so that each HP_UNITDATA.indication causes exactly one HP_UNITDATA.request.

6.3.1 HP_UNITDATA.request

This primitive transfers the symbol data stream from P-MAC to H-MUX.

6.3.1.1 Semantics of the primitive

HP_UNITDATA.request (
HP_Request(symbol)
)

The symbol specified by HP_Request (symbol) shall be one of the following:

I, J, K, L, n, R, S or T, from the set of symbols defined in the FDDI standard on PHY.

6.3.1.2 When generated

P-MAC shall generate one HP_UNITDATA.request for each HP_UNITDATA.indication received from H-MUX.

6.3.1.3 Effect of receipt

Upon receipt of this primitive the H-MUX shall accept the symbol and multiplex it into the transmit data stream in the Packet Data Channel.

6.3.2 HP_UNITDATA.indication

This primitive transfers the symbol data stream to P-MAC from H-MUX.

6.3.2.1 Semantics of the primitive

```
HP_UNITDATA.indication          (
                                HP_Indication(symbol)
                                )
```

The symbol specified by HP_Indication (symbol) shall be one of the following:

H, I, J, K, L, n, R, S, T and optionally Q or V, from the set of symbols defined in ISO/IEC 9314-1.

6.3.2.2 When generated

H-MUX shall generate a HP_UNITDATA.indication for each symbol received from PHY in basic mode, and for each packet data channel symbol in hybrid mode. This indication shall be generated once every symbol period in basic mode. Since the packet data channel is byte interleaved with other information in hybrid mode, P-MAC shall be capable of processing intermittent pairs of indications in hybrid mode.

6.3.2.3 Effect of receipt

Upon receipt of this primitive, MAC shall accept a symbol from H-MUX, process it, and generate a corresponding HP_UNITDATA.request to H-MUX, conveying the resulting output symbol.

6.3.3 HP_INVALID.indication

This primitive indicates to P-MAC that continuity of the received symbol stream has been compromised.

6.3.3.1 Semantics of the primitive

```
HP_INVALID.indication           (
                                HP_Invalid
                                )
```

The HP_Invalid parameter shall indicate that the HP_UNITDATA.indication symbol stream is invalid.

6.3.3.2 When generated

H-MUX shall generate this primitive whenever it does not have a valid symbol stream to present to P-MAC.

6.3.3.3 Effect of receipt

The effect of receipt of this primitive by P-MAC is unspecified.

6.3.4 HP_MODE.request

The function of this primitive is to allow the P-MAC to force transmission of MAC recovery frames (*e.g.* Claim and Beacon) in basic mode.

6.3.4.1 Semantics of the primitive

HP_MODE.request (
HP_mode
)

The HP_mode parameter shall be set to 'any' to allow normal hybrid operation and to 'basic' to force transmission of frames in basic mode.

6.3.4.2 When generated

P-MAC continuously sends an indication of its mode to the local H-MUX entitiy. P-MAC shall allow normal H-MUX operation except during transmission of MAC frames during ring recovery.

6.3.4.3 Effect of receipt

When HP_mode(any) is asserted the H-MUX is free to transmit cycles in the normal manner. When the HP_mode(basic) is asserted the H-MUX shall transmit MAC SDUs in basic mode.

Only the channel exchange process is affected by this primitive. All other parts of H-MUX are unaffected and continue to operate normally.

6.4 H-MUX to I-MAC services

This subclause specifies the services provided at the interface between the I-MAC and H-MUX entities to allow I-MAC to exchange I-MAC protocol data units with peer I-MAC entities.

There are 16 HI-SAPs defined at the H-MUX to I-MAC interface, corresponding to the total of 16 WBCs available for isochronous applications. When the ring operates in basic mode, all the HI-SAPs are closed. When the ring is in hybrid mode, the SMT opens and closes the appropriate HI-SAPs in response to the WBC Assignment process. The definition of 16 separate HI-SAPs is not intended to imply any particular physical implementation.

The following primitives are defined:

HI_UNITDATA.request

HI_UNITDATA.indication

HI_INVALID.indication

HI_CYCLE_SYNC.indication

Each primitive includes the information that is passed between the I-MAC and H-MUX entities. All H-MUX data service primitives have the duration of an integral number of symbol periods.

6.4.1 HI_UNITDATA.request

This primitive defines the transfer of data from I-MAC to the local H-MUX entity.

6.4.1.1 Semantics of the primitive

HI_UNITDATA.request (
HI_Request (HI-SAP,symbol)
)

The symbol specified by HI_Request (HI-SAP,symbol) shall be either I or n, where n is any of the 16 data symbols specified in PHY.

6.4.1.2 When generated

I-MAC shall send H-MUX a HI_UNITDATA.request following receipt of each HI_UNITDATA.indication primitive for that HI-SAP. The time delay between corresponding HI_UNITDATA.indication and HI_UNITDATA.request primitives is implementation dependent and may be null. The timing of the HI_UNITDATA.request primitive shall be such that during normal operation 192 HI_UNITDATA.request primitives, corresponding to the received indication primitives, (comprising the data for one WBC for one cycle) are issued for each open HI-SAP between each assertion of the HI_CYCLE_SYNC.indication primitive.

6.4.1.3 Effect of receipt

Upon receipt of this primitive the H-MUX shall accept the symbol and multiplex it into the transmit data stream at the appropriate point in time.

6.4.1.4 Additional comments

During normal operation the timing of the HI_CYCLE_SYNC.indication primitive will be regular, and define the start of cycle, however the precise timing of the HI_UNITDATA.request primitives can be affected by the preamble and cycle header.

6.4.2 HI_UNITDATA.indication

This primitive defines the transfer of data from the H-MUX entity to I-MAC.

6.4.2.1 Semantics of the primitive

HI_UNITDATA.indication (
HI_Indication (HI-SAP,symbol)
)

The symbol specified by HI_Indication (HI-SAP,symbol) shall be one of the following: I, L, n, T, R, S, and optionally Q, H, or V.

6.4.2.2 When generated

The H-MUX entity sends I-MAC an HI_UNITDATA.indication whenever H-MUX has a valid I-MAC symbol (as defined in 6.3.2.1) from the receive data stream available for output and the appropriate HI-SAP is open. The timing of the HI_UNITDATA.indication primitive shall be such that during normal operation 192 HI_UNITDATA.indication primitives are issued for each open HI-SAP between each assertion of the HI_CYCLE_SYNC.indication primitive. This indication is never sent in basic mode.

6.4.2.3 Effect of receipt

Upon receipt of this primitive, I-MAC shall accept a symbol from H-MUX .

6.4.2.4 Additional comments

During normal operation the timing of the HI_CYCLE_SYNC.indication primitive will be regular, and define the start of cycle, however the precise timing of the HI_UNITDATA.indication primitives can be affected by the preamble and cycle header.

6.4.3 HI_INVALID.indication

This primitive is generated by the H-MUX and asserted to I-MAC to indicate that H-MUX is unable to present a valid data stream to I-MAC.

6.4.3.1 Semantics of the primitive

```
HI_INVALID.indication          (
                               HI_Invalid(HI-SAP)
                               )
```

The HI_Invalid parameter shall indicate that H-MUX is unable to present a valid data stream to I-MAC.

6.4.3.2 When generated

The H-MUX generates this primitive whenever the receiver does not have a valid I-MAC data symbol (as defined in 6.3.2.1) to output for this HI-SAP. This may be due to the associated HI-SAP being closed, the receipt of an illegal symbol at the PHY interface, or the receipt of a PH_INVALID.indication at the PHY interface.

6.4.3.3 Effect of receipt

The correct course of action on receipt of invalid data in an isochronous channel is application dependent. Therefore upon receipt of this primitive, I-MAC shall pass a IM_INVALID.indication to CS-MUX in the appropriate channels.

6.4.4 HI_CYCLE_SYNC.indication

The function of this primitive is to provide 8 kHz timing and cycle sequence information to I-MAC for use in timing generation for isochronous data.

6.4.4.1 Semantics of the primitive

```
HI_CYCLE_SYNC.indication       (
                               HI_cycle_sync,
                               HI_cycle_sequence
                               )
```

The HI_cycle_sync parameter shall be used to provide 8 kHz timing to I-MAC.

The HI_cycle_sequence parameter shall be used to provide the cycle sequence number to I-MAC.

6.4.4.2 When generated

During normal synchronized operation in hybrid mode H-MUX shall send HI_CYCLE_SYNC.indication to I-MAC at 125 µs intervals as defined by the 8 kHz timing used for transmission of hybrid cycles. During hybrid synchronization the time intervals between the start of cycle transmission can vary due to resynchronisation to a received cycle: this will be reflected in the periodicity of the HI_CYCLE_SYNC.indication primitive. Provision of this primitive in basic mode is optional, if it is provided a local frequency reference may be used.

6.4.4.3 Effect of receipt

I-MAC shall use this primitive to generate the 8 kHz timing for isochronous data.

6.4.4.4 Additional comments

Data transfer timing between I-MAC and H-MUX is referenced to the HI_CYCLE_SYNC.indication primitive. In a cycle master the timing of this primitive is phase locked to the phase of the SM_HM_CYCLE_SYNC.request primitive input to H-MUX. However no predetermined phase between the two can be assumed.

6.5 H-MUX to SMT services

This subclause specifies the services provided at the interface between the station management (SMT) entity and H-MUX. This interface is used by the local SMT entity to monitor and control the operation of H-MUX. Additional detail is provided in SMT concerning conditions that generate these primitives and SMT actions upon receipt of H-MUX-generated primitives.

The following primitives are defined:

SM_HM_INITIALIZE_PROTOCOL.request

SM_HM_CONTROL.request

SM_HM_STATUS.indication

SM_HM_SAP.request

SM_HM_WBC.request

SM_HM_WBC.confirm

SM_HM_CYCLE_SYNC.request

Each primitive includes the information that is passed between H-MUX and SMT.

All primitives described in this subclause are mandatory in stations that support the optional monitor functions. Primitives and parameters pertaining solely to the optional monitor functions need not be supported by non-monitor stations.

6.5.1 SM_HM_INITIALIZE_PROTOCOL.request

This primitive has local significance and is used by SMT to change operational parameters of H-MUX. SMT may change any or all of these parameters at any time regardless of whether H-MUX is operational in the ring; however a change in some of the parameters may not be within the domain of normal FDDI-II operation.

6.5.1.1 Semantics of the primitive

```
SM_HM_INITIALIZE_PROTOCOL.request    (
                                     Rank_value,
                                     Initialize_flag,
                                     Sustain_flag,
                                     Maintain_flag
                                     Contend_flag,
                                     T_Hold_value,
                                     T_Valid_value,
                                     T_Err_value,
                                     T_Recov_value,
```

PA_Max,
)

The Rank_value parameter sets the value of the station monitor rank. The rank specifies the current negotiated monitor precedence (default 0).

The Initialize_flag parameter sets the value of the Initialize flag. This flag is set true by SMT if it intends the ring to operate in hybrid mode and it wishes to allow this station to take the ring to hybrid mode on TVS expiry.

The Sustain_flag parameter sets the value of the Sustain flag. This flag is set true by SMT in a monitor if it wishes to allow the HRC entity to try to maintain hybrid mode synchronization when it detects a cycle error by continuing to source cycles.

The Maintain_flag parameter sets the value of the Maintain flag. This flag is set true by SMT in a monitor if it wishes to allow cycle master resolution in hybrid mode when two active rings are merged.

The Contend_flag parameter sets the value of the Contend flag. This flag is set true by SMT when it requires cycle control to go to the STANDBY state, from BASIC or SLAVE states, on receipt of JKRR[<rank] (i.e. a contention cycle header with a rank lower than own rank). This parameter is set true by SMT in all monitors on the ring if a requirement exists for the highest ranked monitor to always become cycle master.

The T_Hold_value parameter sets the T_Hold timeout value.

The T_Valid_value parameter sets the T_Valid timeout value.

The T_Err_value parameter sets the T_Err timeout value.

The T_Recov_value parameter sets the T_Recov timeout value.

The PA_max parameter specifies the maximum number of symbol periods between the start of a preamble and the point at which a cycle error would be signalled to cycle control if the subsequent cycle header were not received. After PA_max symbol periods without receiving a cycle, a monitor with the sustain flag set will begin to source a new cycle.

The value of PA_max depends on the degree of internal smoothing available in H-MUX. The default corresponds to the length of the maximum allowed preamble (6), plus two symbols allowance for an Elasticity Buffer quantum if the required PHY SDU Filter smoothing capacity at the maximum preamble threshold is implemented after H-MUX in the station repeat path.

All parameters of this primitive are optional. If a parameter is omitted, H-MUX shall use the most recently provided value for this parameter or, if no value has previously been provided, the default value for the parameter.

6.5.1.2 When generated

This primitive is generated by SMT whenever it requires H-MUX to reconfigure.

6.5.1.3 Effect of receipt

Receipt of this primitive shall cause H-MUX to establish the values of its address, timers, and other initialization parameters. Upon completion of this primitive, H-MUX shall generate a SM_HM_INITIALIZE_PROTOCOL.confirm.

6.5.2 SM_HM_CONTROL.request

This primitive is used by SMT to control the operation of H-MUX.

6.5.2.1 Semantics of the primitive

SM_HM_CONTROL.request	(Control_Action, Class_value Requested_Status, Requested_Start_Mode,)

The Control_Action parameter shall include the following: Set_Class, HRC_Reset, HRC_Start, HRC_Stop, Present_Status, Begin_Loopback_Mode, or Cancel_Loopback_Mode.

The Class_value parameter specifies the value of the Class variable in HRC (default SLAVE).

The Requested_Status parameter is associated with the Present_Status action and shall indicate the status to be presented.

The Requested_Start_Mode parameter is associated with the HRC_Start action and shall take a value of either "master" or "contend". The significance of these parameters is defined in cycle control.

6.5.2.2 When generated

This primitive is generated by SMT to cause H-MUX to take the action specified by the Control_Action parameter.

6.5.2.3 Effect of receipt

If the control_action is Set_Class then the Class variable shall be set to Class_value (default SLAVE).

If the Control_Action is Reset then H-MUX shall generate the HRC_Reset signal. This shall cause all state machines, timers and counters to be reset to their initial values.

If the Control_Action is HRC_start then if H-MUX is in basic mode it shall make a transition to hybrid mode and attempt to effect hybrid synchronization. If H-MUX is in hybrid mode it shall ignore this primitive.

If the Control_Action is HRC_stop then if H-MUX is in hybrid mode it shall make a transition to basic mode. If H-MUX is in basic mode it shall ignore this primitive.

If the Control_Action is Present_Status, then H-MUX shall generate a SM_HM_STATUS.indication to SMT that includes the status indicated by the Requested_Status parameter.

If the Control_Action is Begin_Loopback_Mode, then H-MUX shall enter Loopback mode. The intent of this mode is to loop back within the H-MUX entity at a point as close as possible to the interface with PHY to permit local station testing. In this mode, H-MUX shall return symbols presented at the HP_UNITDATA.request interface on the HP_UNITDATA.indication interface. H-MUX shall present continuous Idle symbols to the PH_UNITDATA.request interface while in Loopback mode.

If the Control_Action is Cancel_Loopback_Mode, then H-MUX shall leave Loopback mode.

6.5.3 SM_HM_STATUS.indication

This primitive is generated by H-MUX to inform SMT of requested current status, or errors and significant status changes detected by H-MUX.

6.5.3.1 Semantics of the primitives

SM_HM_STATUS.indication	(status_report)

The status_report parameter shall convey the appropriate status including the following:

a) Current state of H-MUX including the current value of

Class variable
cycle control state
CCopr
CSopr
Valid
template filter states
Template Generate states (for monitors)

Optionally the current value of

C_Flag
CCrc
CSrc and CSexp
Scrub
Generate

b) Any change in the values of: Class, cycle control state, CCopr, or CSopr (other than setting CSopr = next(CSopr), template filter state or Template Generate state (for monitors).

c) Failure to establish or maintain successful operation in the slave, Standby or Resynch states (return to basic mode due to TNS expiration).

d) Receipt of a cycle header with potential duplicate rank.

e) The current value of a counter when requested, including: Received_ct, Aborted_ct, Late_ct, Sync_Error_ct, Seq_Error_ct, Header_Error_ct, Generate_ct.

f) Increment of a counter.

g) Overflow of a counter (optional).

6.5.3.2 When generated

These primitives shall be generated by H-MUX to signal the occurrence of a reportable condition, or in response to a SM_HM_CONTROL.request(present_status).

6.5.3.3 Effect of receipt

The effect of receipt of this primitive by SMT is not specified.

6.5.4 SM_HM_SAP.request

This primitive is used by SMT to change the Open template for the HI-SAPs and the HP-SAP within H-MUX.

6.5.4.1 Semantics of the primitive

```
SM_HM_SAP.request          (
                           Control_Action,
                           Requested_SAP
                           )
```

The Control_Action parameter shall include the following: Open_SAP, Close_SAP.

The Requested_SAP parameter selects the HI-SAP or HP-SAP to be modified by the Control_Action.

6.5.4.2 When generated

This primitive is generated by SMT whenever it requires H-MUX to reconfigure its Open[HI-SAP] or Open[HP-SAP] template information. SMT may request a change in the Open template at any time; however, the Open[HI-SAP] template is only used when H-MUX is operational in a hybrid ring.

6.5.4.3 Effect of receipt

The initial and default condition is all SAPs closed.

Receipt of this primitive shall cause H-MUX to modify the Open[HI-SAP] or Open[HP-SAP] template as specified by the Requested_SAP parameter.

If the Requested_SAP parameter is HI-SAP[n], the selected HI-SAP shall be opened or closed to the local I-MAC.

If the Requested_SAP parameter is HP-SAP, the local HP-SAP shall be opened or closed to the local P-MAC.

6.5.5 SM_HM_WBC.request

This primitive is used by SMT to change the Asg[WBC] template within H-MUX. This template exists in each monitor H-MUX; however, only the template in the cycle master affects cycle programming.

6.5.5.1 Semantics of the primitive

```
SM_HM_WBC.request          (
                           Control_Action,
                           Requested_WBC
                           )
```

The Control_Action parameter shall include the following: Assign_Packet, Assign_Isochronous.

The Requested_WBC parameter selects the WBC to be modified by the Control_Action.

6.5.5.2 When generated

This primitive is generated by SMT whenever it requires H-MUX to reconfigure its Asg[WBC] template information. SMT may request a change in the programming template at any time; however the change can only be effected when H-MUX is the cycle master in a hybrid ring.

6.5.5.3 Effect of receipt

The initial and default condition of the Asg[WBC] is TK_request.

Receipt of this primitive shall cause H-MUX to modify the Asg[WBC] template in accordance with the parameters of the primitive. Upon completion of this primitive, H-MUX shall generate a SM_HM_WBC.confirm.

If the Control_Action is Assign_Isochronous, H-MUX shall set the Asg[WBC] flag corresponding to the WBC specified in the Requested_WBC parameter to ISO_request.

If the Control_Action is Assign_Packet, H-MUX shall set the Asg[WBC] flag corresponding to the WBC specified in the Requested_WBC parameter to TK_request.

6.5.6 SM_HM_WBC.confirm

This primitive is used by the H-MUX to inform SMT that the SM_HM_WBC.request is complete.

6.5.6.1 Semantics of the primitive

SM_HM_WBC.confirm (
status
)

The status parameter indicates the success or failure of the SM_HM_WBC.request.

6.5.6.2 When generated

This primitive shall be generated by H-MUX upon completion of a SM_HM_WBC.request. If the H-MUX is currently the cycle master (cycle control state is Resynch or master), then this primitive shall be generated when the new template is received from the ring by H-MUX. Otherwise, this primitive shall be generated when the Asg[WBC] template has been updated.

6.5.6.3 Effect of receipt

The effect of receipt of this primitive by SMT is not specified.

6.5.7 SM_HM_CYCLE_SYNC.request

The function of this primitive is to provide 8 kHz timing information to H-MUX for use in hybrid cycle timing generation. This primitive is only used by H-MUX in a cycle master station. The capability to support this primitive is mandatory, however its provision by SMT is optional, and may depend on the availability of a frequency reference.

6.5.7.1 Semantics of the primitive

SM_HM_CYCLE_SYNC.request (
cycle_sync
)

The cycle_sync parameter shall be used to provide an 8 kHz cycle clock to H-MUX. This clock may be created internally within the H-MUX implementation or supplied to H-MUX.

Characteristics of the cycle clock shall be:

a) Nominal symbol time (UI) = 40 ns (1/UI = 25 MHz)

b) Nominal cycle time = 125 μs (microseconds)

c) Frequency accuracy < ±0,005 % (±50 ppm)

d) Phase jitter (above 4 kHz) < ±8 ns (0,4 UI pp)

e) Phase jitter (below 4 kHz) < ±240 ns (12 UI pp)

6.5.7.2 When generated

SMT sends SM_HM_CYCLE_SYNC.request to H-MUX at 125 μs intervals as defined by the master 8 kHz reference used for isochronous data within the station. This 8 kHz reference may be local or it may have been derived from some external reference such as the public digital network. If an external reference is used the signal can on occasion be lost, in which case this primitive may not be sent for a period of time until recovery of the external reference has been effected.

6.5.7.3 Effect of receipt

In a cycle master the receipt of this primitive shall cause H-MUX to lock the phase of its local 8 kHz timing to the external reference. The actual (locked) phase between the input reference, specified by the

SM_HM_CYCLE_SYNC.request primitive, and the phase of transmitted cycles is not specified. The phase may also depend on implementation requirements. In a slave station this primitive is ignored.

6.5.7.4 Additional comments

In the cycle master mode, H-MUX should be designed to tolerate both the absence of this primitive, and the receipt of this primitive following a period of absence, without any phase jumps in the cycles transmitted on the ring. If a phase jump occurs, cycle control should return to the Resynch state to resynchronize the ring.

6.6 I-MAC to CS-MUX services

This subclause specifies the services provided at the interface between the I-MAC and CS-MUX entities to allow CS-MUX to exchange CS-MUX protocol data units with peer CS-MUX entities. Since channels vary in size CS-MUX protocol data units may also vary in size.

Based on the steering map provided by SMT, I-MAC selects the currently open channels from the open WBCs delivered by H-MUX and delivers these channels to CS-MUX. In the opposite direction I-MAC receives channels from CS-MUX. The I-SAP (Isochronous service access point) is the access point, at the interface between CS-MUX and I-MAC, used to access channels. There is one I-SAP defined per channel allocated at the I-MAC. When the ring operates in basic mode, all the I-SAPs are closed. When the ring is in hybrid mode, the SMT opens and closes the appropriate I-SAPs in response to the Channel Allocation process.

The following primitives are defined:

IM_UNITDATA.request

IM_UNITDATA.indication

IM_INVALID.indication

IM_CYCLE_SYNC.indication

Each primitive includes the information that is passed between the I-MAC and CS-MUX entities.

6.6.1 IM_UNITDATA.request

This primitive defines the transfer of data from CS-MUX to the local I-MAC entity.

6.6.1.1 Semantics of the primitive

```
IM_UNITDATA.request          (
                             IM_Request (I-SAP,CS-MUX PDU)
                             )
```

The IM_Request (I-SAP,CS-MUX PDU) shall specify a unit of isochronous data associated with the corresponding channel. The size of this unit of data is a function of channel size.

6.6.1.2 When generated

CS-MUX shall send I-MAC an IM_UNITDATA.request following receipt of each IM_UNITDATA.indication primitive for that I-SAP. The time delay between corresponding IM_UNITDATA.indication and IM_UNITDATA.request primitives is implementation dependent and may be null.

6.6.1.3 Effect of receipt

Upon receipt of this primitive the I-MAC shall accept the CS-MUX PDU and multiplex it into the appropriate WBC for transmission to H-MUX.

6.6.2 IM_UNITDATA.indication

This primitive defines the transfer of data from the I-MAC entity to CS-MUX.

6.6.2.1 Semantics of the primitive

```
IM_UNITDATA.indication          (
                                IM_Indication (I-SAP,CS-MUX PDU)
                                )
```

The IM_Indication (I-SAP,CS-MUX PDU) shall specify a unit of isochronous data associated with the corresponding channel. The size of this unit of data is a function of channel size.

6.6.2.2 When generated

The I-MAC entity sends CS-MUX an IM_UNITDATA.indication whenever I-MAC has a CS-MUX PDU from the receive data stream available for output and the appropriate I-SAP is open. The period of time between subsequent IM_UNITDATA.indication generations will vary according to the data rate of the Isochronous channel. In basic mode this indication is never sent.

6.6.2.3 Effect of receipt

Upon receipt of this primitive, CS-MUX shall accept a CS-MUX PDU from I-MAC. Receipt of this primitive by CS-MUX shall also signify acceptance by I-MAC of the CS-MUX PDU specified by the current IM_UNITDATA.request and willingness to accept another CS-MUX PDU.

6.6.3 IM_INVALID.indication

This primitive is generated by the I-MAC and asserted to CS-MUX to indicate that I-MAC is unable to present a valid CS-MUX PDU to CS-MUX.

6.6.3.1 Semantics of the primitive

```
IM_INVALID.indication           (
                                IM_Invalid(I-SAP)
                                )
```

The IM_Invalid parameter shall indicate that I-MAC is unable to present a valid CS-MUX PDU stream to CS-MUX.

6.6.3.2 When generated

The I-MAC generates this primitive whenever it does not have a valid CS-MUX PDU to output to CS-MUX for this I-SAP. This may be due to the associated I-SAP being closed or to the receipt of a HI_INVALID.indication at the H-MUX interface.

6.6.3.3 Effect of receipt

The effect of receipt of this primitive on CS-MUX is not specified.

6.6.4 IM_CYCLE_SYNC.indication

The function of this primitive is to provide 8 kHz and cycle sequence information to CS-MUX for use in timing generation for isochronous data.

6.6.4.1 Semantics of the primitive

```
IM_CYCLE_SYNC.indication        (
                                IM_cycle_sync
                                IM_cycle_sequence
                                )
```

The IM_cycle_sync parameter shall be used to provide 8 kHz timing to CS-MUX.

The IM_cycle_sequence parameter shall be used to provide the cycle sequence number to CS-MUX.

6.6.4.2 When generated

During normal synchronized operation in hybrid mode, I-MAC shall send IM_CYCLE_SYNC.indication to CS-MUX at 125 µs intervals as defined by the 8 kHz timing defined by the HI_CYCLE_SYNC.indication primitive received from H-MUX. During hybrid synchronization, the time intervals between HI_CYCLE_SYNC.indication primitives can vary due to resynchronisation; this will be reflected in the periodicity of the IM_CYCLE_SYNC.indication primitive. Provision of this primitive in basic mode is optional.

6.6.4.3 Effect of receipt

CS-MUX shall use this primitive to generate the 8 kHz timing for isochronous data.

6.6.4.4 Additional comments

Data transfer timing between CS-MUX and I-MAC is referenced to the IM_CYCLE_SYNC.indication primitive. In a cycle master the timing of this primitive is phase locked to the phase of the HI_CYCLE_SYNC.indication primitive output from H-MUX.

6.7 I-MAC to SMT services

This subclause specifies the services provided at the interface between the station management (SMT) entity and I-MAC. This interface is used by the local SMT entity to monitor and control the operation of I-MAC. Additional detail is provided in SMT concerning conditions that generate these primitives and SMT actions upon receipt of I-MAC-generated primitives.

The following primitives are defined:

SM_IM_CONTROL.request

SM_IM_SAP.request

All primitives described in this clause are mandatory. Each primitive includes the information that is passed between I-MAC and SMT.

6.7.1 SM_IM_CONTROL.request

This primitive has local significance and is used by SMT to control the operation of I-MAC.

6.7.1.1 Semantics of the primitive

SM_IM_CONTROL.request (
Control_Action,
Requested_Status
)

The Control_Action parameter shall include the following: Reset, and Present_Status.

The Requested_Status parameter shall include the following: current value of the status of any channel, I-MAC statistics.

6.7.1.2 When generated

This primitive is generated by SMT to cause I-MAC to take the action specified by the Control_Action parameter.

6.7.1.3 Effect of receipt

If the Control_Action is Reset then I-MAC shall be reset to its initial state with all channels closed.

If the Control_Action is Present_Status, then I-MAC shall present the status to SMT as indicated by the Requested_Status parameter.

6.7.2 SM_IM_SAP.request

This primitive is used by SMT to change the channel steering map within I-MAC. SMT may request a change in the channel steering map at any time regardless of whether or not I-MAC is operational in the ring.

6.7.2.1 Semantics of the primitive

SM_IM_SAP.request (
Control_Action,
Requested_Channel,
)

The Control_Action parameter shall include the following: Open Channel, Close Channel.

The Requested_Channel parameter shall specify the size of the channel, the WBC in which the channel is assigned, and its location within the WBC (see 5 for the size of allowed channels). Additional associated channel descriptors, such as the optional channel security label, are also specified by this parameter.

6.7.2.2 When generated

This primitive is generated by SMT whenever it requires I-MAC to reconfigure its channel steering map.

6.7.2.3 Effect of receipt

The initial and default condition is that all I-SAPs are closed.

Receipt of this primitive shall cause I-MAC to modify the channel steering map in accordance with the parameters of the primitive.

If the Control_Action is Open Channel the channel specified in the Requested_Channel parameter shall be opened by I-MAC. Overlapping channels may not be opened, the effect of attempting to open overlapping channels is not defined.

If the Control_Action is Close Channel the channel specified in the Requested_Channel parameter shall be closed by I-MAC.

7 Facilities

7.1 H-MUX symbol set

The H-MUX uses the same symbol set as defined for the FDDI PHY-2 and MAC-2. This is composed of the: I, J, K, L, n (0-F), R, S and T symbols, and optionally Q, H and V. In basic mode, all Physical Indication symbols are delivered to the P-MAC as HP_IND.

7.2 Cycle

The Cycle is the HRC frame that carries packet and isochronous data. A new cycle is generated by the cycle master every 125 µs, and cycles which have been repeated around the ring are stripped by the cycle master. The length of the cycle is dependent on the transmission rate of the ring. All descriptions in this clause assume a transmission rate of 100 Mbps. At 100 Mbps, the cycle is 1 560 symbol pairs, plus the preamble. Each cycle is recognized by a hybrid multiplexer at every station. There can be more than one cycle circulating around the ring at any one point in time, depending on the ring size.

This Cycle is partitioned into four parts: the preamble, the cycle header, the dedicated packet group, and the WBC cyclic groups as shown in figure 9.

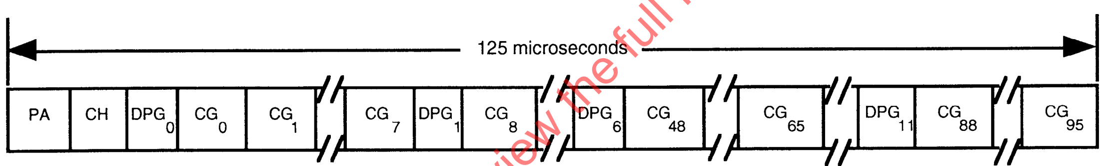


Figure 9 — H-MUX cycle structure at 100 Mbps

Cycle: 125 µs (nominally 3 125 symbols)

PA: preamble (nominally 5 symbols)

CH: cycle header (24 symbols)

DPG0 -11: dedicated packet group (24 data symbols). The DPG is byte interleaved among the cyclic groups

CG0-95: cyclic group 0 to 95 (32 symbols, 2 symbols per WBC). The n^{th} symbol pair in each cyclic group belongs to the n^{th} WBC, where *n* ranges from 0 to 15.

7.2.1 Preamble (PA)

The preamble is nominally 5 symbols long. The cycle clocking algorithm will normally produce preamble lengths between 4 and 6 symbols.

7.2.2 Cycle header (CH)

The cycle header establishes the 125 µs boundary and conveys the cycle control information and the programming template. The programming template identifies each of the sixteen 6,144 Mbps WBCs for either isochronous or packet data transmission. Isochronous WBCs are individually usable by the I-MAC. The packet data WBCs are combined with the DPG to form the packet data channel (PDC). The PDC is used by the P-MAC. Figure 10 illustrates the format of the cycle header.

SD start delimiter	C1 synch control	C2 seq control	CS cycle sequence	P0	P1	programming template	P15	IMC isochronous maint chan

Figure 10 — H-MUX cycle header

SD: starting delimiter (symbols J and K).

C1: synchronization control (1 symbol). R and S are the permitted values.

C2: sequence control (1 symbol). R and S are the permitted values.

CS: cycle sequence (2 symbols).

P0-P15: programming template (16 symbols). R and S are the symbols used to indicate whether each WBC is carrying packet data (R symbol) or isochronous data (S symbol). The symbol T is substituted for a corrupted R or S by the station detecting the error.

IMC: isochronous maintenance channel (2 symbols).

7.2.2.1 Starting delimiter (SD)

The starting delimiter indicates the beginning of a cycle. SD is represented by the JK symbol pair which forms a unique ten bit sequence.

7.2.2.2 Synchronization control (C1)

The synchronization control symbol, C1, is used to establish the synchronization state of the ring: it may take the value R or S.

A value of R indicates that synchronization has not been established and any cycle may be legitimately interrupted by a new cycle.

A value of S indicates that synchronization has been established and cycle hold mode may be entered meaning that the current cycle may not be legitimately interrupted by a new cycle. In normal operation, C1 is expected to be set to S.

C1 may be set to S only by the cycle master, slave stations may either repeat the received value of C1 or reset it to R. If any station, other than the cycle master, receives C1 with a value not equal to S, it will set C1 to R. A cycle master sets C1 to S once full synchronization has been achieved. Slave stations detecting a cycle synchronization error reset C1 to R. As slave stations may never set C1, so synchronization errors on the ring are always indicated to the cycle master by the receipt of C1 = R.

Immediately after switching to hybrid mode or after an Error Recovery period, stations on the ring might not receive cycles periodically (i.e. every 125 μs). The cycle master initiates the synchronization process by transmitting cycles continuously with C1 values set to R. When the transmitted cycles return to the cycle master, this indicates that all stations on the ring are receiving and repeating cycles successfully. The cycle master then transmits cycles with C1 set to S.

During normal operation, if a monitor station does not receive a cycle within 125 μs of the previous cycle, it may start generating cycles with C1 set to R. In the same situation, a non-monitor station will change C1 to R in the next repeated cycle. On receipt of a cycle, it resets its 125 μs timer to wait for the next cycle. The C1 value of R indicates that synchronization has been lost. Since slave stations never set C1 any synchronization errors on the ring are always indicated to the cycle master by the receipt of C1 = R.

7.2.2.3 Sequence control (C2)

The sequence control symbol is used for loading and latching the cycle sequence (CS): it may take the value R or S.

A value of R in the C2 field indicates that either the cycle sequence has not yet been established or that a cycle sequence error has been detected.

A value of S in the C2 field indicates that valid cycle sequence is established and stations can latch each CS value to compare with the CS value of the next cycle. During normal operation, the C2 field will be set to S.

C2 may be set to S only by the cycle master, slave stations may either repeat the received value of C1 or reset it to R. If any station, other than the cycle master, receives C2 with a value not equal to S, it will reset C2 to R. During normal hybrid mode operation the cycle master transmits cycles continuously with incremental CS values and sets the C2 field to S to indicate that the CS field contains a valid cycle sequence number. When the cycle master receives cycles with C2 values set to S, it knows that all stations are receiving cycles with sequential CS values. If a slave station does not receive cycles sequentially, it will change C2 to R in the cycle with the detected sequence error. Since slave stations never set C2 any sequence errors on the ring are always indicated to the cycle master by the receipt of C2 = R.

7.2.2.4 Cycle sequence (CS)

The CS field takes the form (NN) where N is a data symbol. Depending on its value it either represents the cycle sequence or a monitor rank.

cycle sequence values from 0 to 63 are reserved for use in the monitor contention procedure. During the Contention process, monitors transmit their rank in the CS field, the monitor with the highest rank becoming the next cycle master.

A CS value of 0 indicates a NULL monitor rank. A station with this rank can not win the monitor contention procedure, however it may initiate it.

A CS value between 1 and 63 indicates a station's monitor rank. The rank is only valid if C1=C2=R, otherwise it must be treated as an error and ignored.

A CS value between 64 and 255 indicates the cyclic sequence in the form: 64 + (n mod 192). Each cycle generated by the cycle master has its CS value incremented by one. The value 255, when incremented, wraps around to 64.

7.2.2.5 Programming template (P0-P15)

The programming template is comprised of 16 symbols which represent the 16 WBCs. Based on the programming template symbol value, the corresponding WBC is used as an isochronous WBC or as part of the PDC.

7.2.2.6 Isochronous maintenance channel (IMC)

The isochronous maintenance channel is dedicated to carry isochronous traffic for maintenance purposes. The recommended use for this channel is to carry voice traffic encoded to the applicable national standard.

7.2.3 Dedicated packet group (DPG)

The dedicated packet group provides a guaranteed minimum bandwidth of 0,768 Mbps (12 bytes of the cycle) for the Packet Data Channel. The octets of the DPG are interleaved with the cyclic groups; the first octet preceding the first cyclic group and succeeding octets occurring after every subsequent eighth cyclic group (see figure 2).

7.2.4 Cyclic groups

The HI_CYCLE_SYNC.indication primitive is asserted by the H-MUX to indicate the beginning of the cyclic groups.

The 16 WBCs within a cycle are byte-interleaved across 96 cyclic groups of 32 symbols each. Each row in figure 2 represents a cyclic group, and contains one byte from each WBC. Since WBCs are numbered from 0 to 15, the byte belonging to WBC *n* is composed of the 2*n* and 2*n*+1 symbol in each cyclic group. Figure 11 illustrates the interleaving of WBC 3 across the cyclic groups. Each WBC carries either packet or isochronous data. Figure 12 illustrates the relationship between the programming template and the WBCs assignment to packet or isochronous mode.

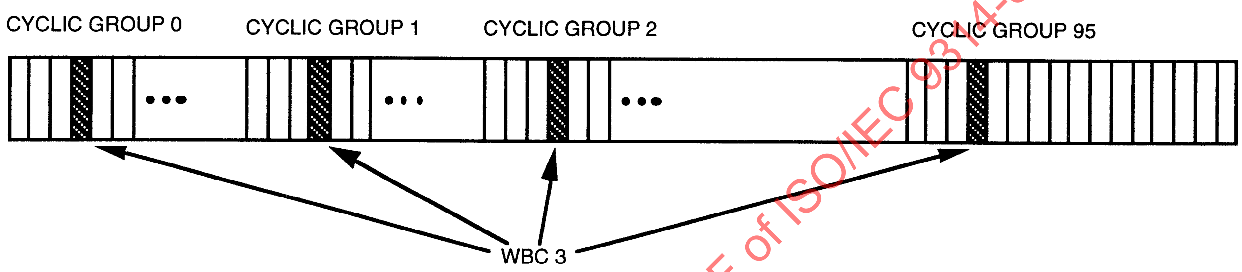


Figure 11 — Example of wideband channel interleaving

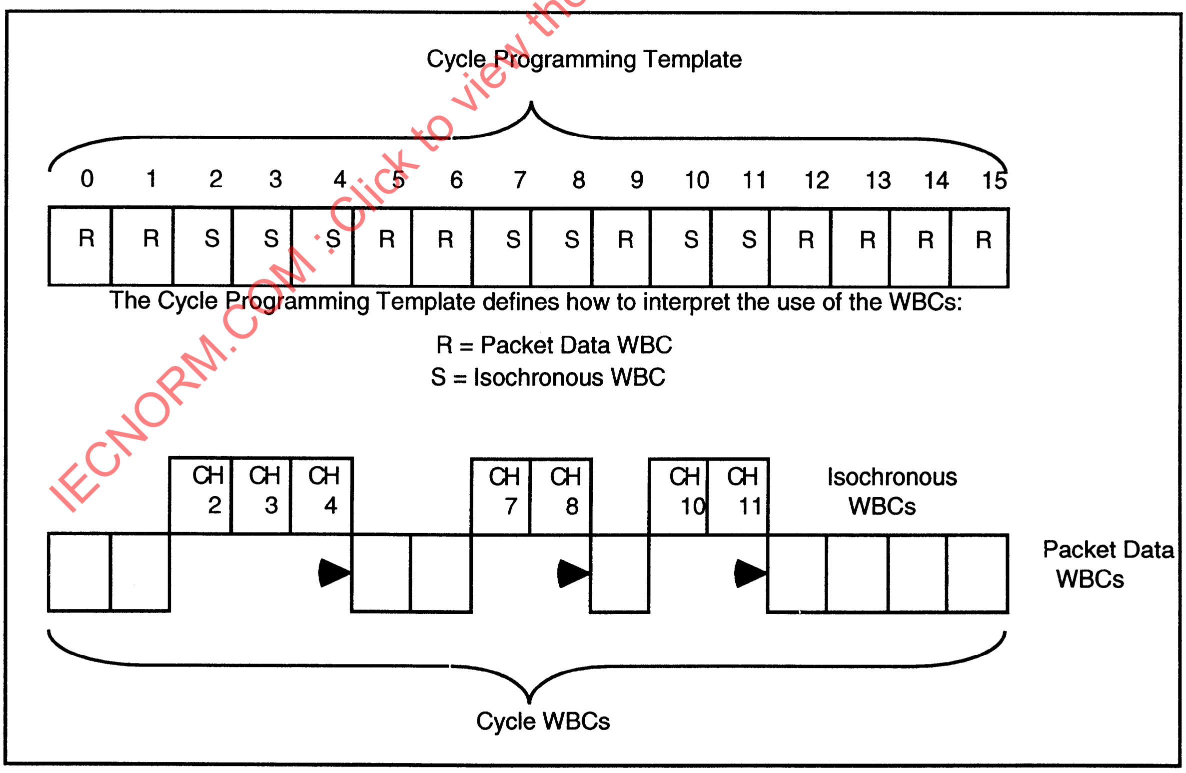


Figure 12 — Example of wideband channel sorting

7.3 HRC protocol parameters

The following parameters control various aspects of the operation of HRC. The values of these parameters would normally be defined by the station management function during station initialization and would only change infrequently during operation.

7.3.1 Parameters required by all stations

All the parameters documented in this clause are mandatory for all stations. It is not generally required that all implementations support the full range of values for each parameter.

Class: This variable is programmed by SMT and the cycle control state machine to control the class of operation of cycle control at this node.

The Class variable may have three values corresponding to three classes of operation:

BASIC

SLAVE

MONITOR

The ability to support the BASIC and SLAVE classes is mandatory.

An H-MUX with the Class variable set to BASIC may only operate in basic mode. The cycle control state machine is forced to remain in the BASIC state.

An H-MUX with the Class variable set to SLAVE may only operate in basic mode or as a slave in hybrid mode.

An H-MUX with the Class variable set to MONITOR may take part in the monitor contention procedure and can become a cycle master. The precise mode of operation is determined by the Initialize, Contend and Sustain flags.

PA_max: The maximum number of symbol periods between the start of a preamble and the point at which a cycle error would be signalled to cycle control if the subsequent cycle header were not received. After PA_max symbol periods without receiving a cycle, a monitor with the sustain flag set will begin to source a new cycle.

The value of this variable depends on the degree of internal smoothing available in H-MUX. The default corresponds to the length of the maximum allowed preamble (6), plus two symbols allowance for an Elasticity Buffer quantum if the required PHY SDU Filter smoothing capacity at the maximum preamble threshold is implemented after H-MUX in the station repeat path.

Open[ISAP]: This is a set of 16 flags which contain a map of which HI-SAPs are open to isochronous traffic at this station. These flags may be set or cleared by SMT using the SM_HM_WBC.request primitive.

Open[PSAP]: This flag indicates whether a P-MAC is currently configured and available at the HP-SAP.

The following parameters are used for the purpose of timer calculation. The values chosen for these parameters have a significant effect on ring operation. If the values chosen are larger than required then ring recovery time will be extended unnecessarily. If the values chosen are too small then, in the worst case, ring synchronization may not be possible. The default values given allow for a wide variety of topologies, and operating conditions.

C_max: Maximum number of cycles which may be contained in a ring (default 23).

C_err: Minimum number of consecutive cycles for which cycle sequence errors are tolerated before some form of recovery is attempted (default 3).

The above parameters are used in the derivation of four time-out values used by the HRC protocol.

T_Hold: This time-out is used during recovery and initialization procedures to ensure that stale cycles and frames are removed from the ring. It is defined as the maximum time a cycle can take to go all round the ring and return to the sending station. It consists of the total ring cable delay plus the maximum total station latency of all stations. The value of this parameter shall be a length of time greater than (C_max) cycle times.

T_Valid: This parameter sets the minimum length of time a station will tolerate cycle sequence errors (allowing for a rotation round the ring) before some form of recovery is attempted. The value of this parameter shall be a length of time greater than (C_max + C_err +2) cycle times.

T_Err: This parameter sets the minimum length of time a station will tolerate cycle sequence errors (not allowing for a rotation round the ring) before some form of recovery is attempted. The value of this parameter shall be a length of time greater than (C_err +2) cycle times.

T_Recov: This parameter is the minimum length of time a station has to spend waiting for cycle sequence synchronization to be established before it may assume that ring synchronization is not currently possible and returns to the BASIC state. The value of this parameter shall be a length of time greater than max(the largest value of T_Valid or (T_Hold + T_Err) on the ring) + C_err + 2.

7.3.2 Parameters required by monitor stations

All the parameters documented in this clause are mandatory for monitor stations. The values of the parameters described are only significant when Class = monitor, and are assumed to be void otherwise.

Rank: Current negotiated monitor precedence. The default value of this variable is 0 indicating that the monitor has no rank and therefore can not win the monitor contention procedure. Any management procedure used to assign other, non-zero, values of this variable must ensure that the values assigned are unique to each HRC entity on the ring.

Initialize: This flag is set true by SMT if it intends the ring to operate in hybrid mode and it wishes to allow this station to take the ring to hybrid mode on TNS expiry.

Sustain: This flag is set true by SMT in a monitor if it wishes to allow the HRC entity to try to maintain hybrid mode synchronization when it detects a missing or aborted cycle by continuing to source cycles.

Maintain This flag is set true by SMT in a monitor if it wishes to allow cycle master resolution in hybrid mode when two active rings are merged.

Contend: This flag is set true by SMT in all monitors on the ring if a requirement exists for the highest ranked monitor to always become cycle master. When this flag is set cycle control always goes to the STANDBY state, from BASIC or SLAVE states, on receipt of JKRR[<rank] (i.e. a contention cycle header with a rank lower than own rank).

Asg[ISAP]: This is a set of 16 flags which contain a map of the current WBC template assignment requests. These flags may be set or cleared by SMT using the SM_HM_WBC.request primitive.

7.4 Variables

The following variables, flags, signals, and timers are used within the H-MUX for the purposes of maintaining status information, providing interfaces between the process, and supporting the operation of the state machines.

A flag is a variable that can take one of two values: set or cleared.

7.4.1 Variables required by all stations

CCrc: Received CC value:- this is the filtered value of the CC field in the last cycle header received. The individual components of this field are C1rc and C2rc.

The C1rc variable can take the values:

R -> Resynchronize

S -> Synchronized

T -> neither R nor S

The C2rc variable can take the values:

R -> Resequence

S -> Sequenced

T -> neither R nor S

CSrc: Received CS value:- this variable is an octet containing the value of the cycle sequence number in the CS field in the last cycle header received. If the received CS field does not contain a valid data symbol pair then the CSrc value is set to zero.

```
IF (received CS field = nn)
   THEN CSrc = (received CS value)
   ELSE CSrc = 0
```

CSexp: Expected CS value:- this is the value of CS the station expects to receive in the next cycle header.

CCopr: Operative CC value:- this is the internal operational value of the cycle control field used by the cycle control process. The individual components of this field are C1opr and C2opr

The C1opr variable can take the values:

R -> Resynchronize

S -> Synchronized

The C2opr variable can take the values:

R -> Resequence

S -> Sequenced

CSopr: Operative CS value:- this is the internal operational value of the cycle sequence number used by the cycle control process. It is used to ensure that cycles are being received in the correct sequence. In a cycle master CSopr is incremented (mod 192) for every cycle received. In SLAVE mode CSopr is overwritten by the cycle sequence number from a received cycle providing the Latch_Sequence condition is satisfied.

PTrc[WBC] A set of 16 variables used to convey the filtered, received WBC template information from the Receive section of the cycle acquisition process. The variables can take the values:

R -> packet WBC

S -> isochronous WBC

T -> neither R nor S

TF[WBC] A set of 16 variables used to convey the state of the WBC Receive state machine from the Receive section of the cycle acquisition process. The variables can take the values:

TK_Initialize

TK_Operate

ISO_Operate

C_flag: This flag is used to define the current modes of operation of the Receive.and cycle generate state machines. A separate copy of this flag exists for each of these state machines. When its C_flag is clear, the state machine is operating in basic mode, when it is set the state machine is operating in hybrid mode. Each state machine is initialized to basic mode. The C_flag is set by the Receive state machine on receipt of a valid cycle header, and by the cycle generate state machine on transmission of a cycle header. Each state machine stays in hybrid mode until its C_flag is cleared.

H_flag This flag indicates that a header error was detected on the current cycle.

M_flag This flag is used to facilitate cycle master resolution in hybrid mode when two active rings are merged. M_flag being set in the RESYNCH state is used as an indication that the state was entered via the MASTER state. When M_flag is set in conjunction with the Maintain flag, the station is allowed to enter monitor contention if it fails to synchronize the ring in the RESYNCH state.

CC_valid: This flag is set by the Receive process on receipt of a valid cycle control field (containing R and S symbols only), and cleared on receipt of an invalid cycle control field.

CS_valid: This flag is set by the Receive process on receipt of a cycle sequence field which is considered valid in the current context. It is cleared on receipt of an invalid cycle sequence field.

Cycle_Error: This flag is set by the Receive process on detection of a cycle synchronization error condition.

Hold: The Hold flag is used in conjunction with the Valid flag and the TNS timer as part of the error recovery process. The Hold flag is normally set by the cycle control state machine whenever TNS is reset to T_Hold or on entry to all states except the MASTER state. It is always reset on TNS expiry. It provides an indication that a state has recently been entered or, in a slave, that a new sequence has recently been started.

Valid: The Valid flag is used in conjunction with the Hold flag and the TNS timer as part of the error recovery process. The Valid flag is normally set by the cycle control state machine when a slave station has received a valid sequence of sequence numbers for longer than T_Hold. It is reset when a new sequence is started.

Scrub: The Scrub flag is used in conjunction with the TVS timer as part of the ring scrubbing process. It is set whenever SMT initiates a transition to hybrid mode to allow HRC to scrub the ring of spurious claim frame headers. Similarly it is set on certain transitions to the BASIC state to allow HRC to scrub the ring of cycle headers. The Scrub flag is always reset on TVS expiry. The Scrub flag is also used to assist in the scrubbing of stale monitor ranks and sequence numbers.

Repeat: The Repeat flag is used by the cycle control state machine to inform the cycle generate state machine when the cycle sequence number passed to it is valid and may be repeated. If it is invalid a null value is substituted.

7.4.2 Variables required only by monitor stations

These variables are required when Class = monitor. When Class = slave CCtx, CStx and PTtx are always equal to CCopr, CSrc and PTrc, respectively.

CCtx: Transmitted CC value:- this is the value of the CC field in the next cycle header to be transmitted. The individual components of this field are C1tx and C2tx and each of these variables can take the value R or S as determined by the cycle generation state machine.

CStx: Transmitted CS value:- this is the value of the CS field in the next cycle header to be transmitted.

CSreq: Requested sequence value:- This variable is used to generate the transmitted cycle sequence number in master mode. CSreq is produced in the Timing Control part of cycle generation and is incremented every time the Cycle_request signal is asserted.

Generate: This flag is set by the cycle generate process when it is generating cycles and not repeating data from the LAB

PTtx[WBC] A set of 16 variables used to convey the operative template for each WBC. The variables can take the values:

R -> packet WBC

S -> isochronous WBC

T -> neither R nor S

TG[WBC] A set of 16 variables used to convey the state of WBC Transmit State Machine in the Cycle Merge section of the channel exchange process. The variables can take the values:

TK_Initialize

TK_Operate

ISO_Initialize

ISO_Operate

7.5 Timers

Each station shall maintain two timers to regulate operation of the hybrid ring. These timers are used in conjunction with the Hold, Valid, and Scrub flags. The significance of timer expiry in any given cycle control state depends on the values of these flags.

For the purpose of this description all timers are assumed to be Reset with the unsigned twos complement of the target time, in either octets or cycles as applicable. Timers are further assumed to count upward if enabled, expiring when an overflow occurs. All timer comparisons are expressed on the basis of elapsed time. These conventions are for reference purposes only and do not prescribe the implementation. Timers run normally in all states if enabled.

TVS: Valid Sequence Timer: This timer is used in conjunction with the Scrub flag as part of the ring scrubbing process. It is reset whenever SMT initiates a transition to hybrid mode to provide a time-out which allows HRC to scrub the ring of spurious claim frame headers. Similarly it is reset on transition to the BASIC state to allow HRC to scrub the ring of cycle headers and also to provide a time-out for return to hybrid mode. The Scrub flag is always reset on TVS expiry. TVS is also used to assist in the scrubbing of stale monitor ranks and sequence numbers and as a time-out for error recovery.

TNS: New Sequence Timer: This timer is used in conjunction with the Hold and Valid flags to control the latching of sequence numbers. It is reset to T_Hold whenever a new sequence number is latched which is not equal to the current CSopr. CSopr latching is then inhibited until expiry. In addition to this use of TNS it is also used as a time-out in ring recovery procedures. TNS is reset on entry to all states except MASTER to provide a time-out from entry to that state.

7.6 Counters

To aid in problem determination and fault location the HRC entity in each station shall maintain the following counts of events in received cycles.

Sync_Error_ct: The number of cycles received with C1rc=S resulting in C1opr being set to R. This is the number of synchronization errors detected at this station.

Seq_Error_ct: The number of cycles received with C2rc=S resulting in C2opr being set to R. This is the number of sequence errors detected by this station.

Header_Error_ct: The number of cycle headers received in error, i.e. received with C1rc≠ R | S or C2rc ≠ R | S or received CS field ≠ nn or a programming template error (symbol ≠ R | S | T).

Aborted_ct: The number of received cycles aborted preemptively before the natural end of the cycle.

Late_ct: The number of times a cycle was not received when expected.

Generate_ct: The number of new cycles transmitted while the Generate flag is set. (monitor stations only)

Received_ct: The number of complete cycles received.

7.7 Signals

A signal is used to initiate a state change within HRC. A signal causes a state change to occur and does not have to be cleared following its usage.

HRC_Start: This signal is used by the SMT to initiate an immediate transition from basic to hybrid mode. A parameter, which may take the values: Contend or master, specifies that the transition is to the STANDBY or RESYNCH states respectively.

HRC_Stop: This signal is used by the SMT to initiate an immediate transition from hybrid to basic mode.

HRC_Reset: This signal is used by the SMT to reset hybrid ring control. SMT may assert this signal at any time. The effect of this signal on cycle control is to cause an immediate transition to the BASIC state. For monitor stations the Class variable is set to SLAVE.

RX_Start: This signal is asserted by the Receive Process on detection of a JK start delimiter.

RX_CC: This signal is asserted by the Receive Process on detection of the cycle control field. During normal hybrid mode operation this consists of the two symbols immediately following the JK start delimiter. The cycle control field symbols are also passed to cycle control as parameters of this signal.

RX_CS: This signal is asserted by the Receive Process on detection of the cycle sequence field. During normal hybrid mode operation this consists of the two symbols immediately following the cycle control field. The cycle sequence field symbols are also passed to cycle control as parameters of this signal. The effect of this signal is to cause cycle control to complete processing of the received cycle header.

RX_MAC: This signal is sent to cycle control by the Receive Process on detection of the start of an FDDI MAC frame as constituted by a JK start delimiter and a frame control field of the form *MN*, where *M* = (8 or C) and *N* = (0 through F). The effect of this signal is to cause cycle control to enter the basic mode of operation providing the Scrub flag is clear.

RX_Beacon: This signal is sent to cycle control by the Receive Process on detection of the start of an FDDI Beacon frame as constituted by a JK start delimiter and a frame control field indicating a Beacon frame. The effect of this signal is to cause cycle control to enter the basic mode of operation regardless of the setting of the Scrub flag.

RX_Abort: This signal is sent by the receive process on any abnormal termination of a cycle. Examples of this include the receipt of a PH_INVALID.indication primitive from PHY, and the situation where the starting delimiter was not in fact part of a cycle.

CC_Backoff: This signal indicates that this station has yielded in the monitor contention procedure. It is asserted by the cycle control state machine on a transition from the Standby to the slave state, and is used by the cycle generate state machine as an indication that it should abort the current cycle and commence transmission of a new one.

TX_Abort: This signal, from the cycle generation process to the channel exchange process, has the effect of causing the channel exchange process to abort transmission of a cycle. .

TX_Start: This signal, from the cycle generation process to the channel exchange process, has the effect of causing the channel exchange process to commence transmission of a cycle. The values of CCtx, CStx and CTtx[WBC] are passed to channel exchange as parameters along with this signal.

Cycle_Request: This signal is sent from the Timing control part of cycle generation to the cycle generate state machine to request the start of a new cycle. The nominal frequency of this signal is 8 kHz.

7.8 Functions

A function operates on the specified input parameters and returns a value

next(CS): This function returns the next value for the cycle sequence field. If CS is less than 64 then it is assumed to be a rank and returned unchanged. If CS is greater than or equal to 64 than the return value is CS incremented mod 192.

```
next(CS):
   IF CS < 64
      THEN  return(CS)
      ELSE     return((((CS - 64) + 1) mod(192)) + 64)
```

8 Operation

8.1 Ring operation overview

This subclause provides a descriptive overview of HRC operation. The formal specifications are contained in subsequent subclauses.

During normal hybrid mode ring operation the cycle master transmits cycles consisting of preamble, cycle header, dedicated packet group, and cyclic data groups every 125 µs. The cycle timing used may be derived from the cycle master's local timing or alternatively it may be derived from some external frequency reference such as ISDN. See annex B for timing considerations. In the absence of errors, slave stations, that is, all stations except the cycle master, monitor and repeat the cycle header unchanged.

The programming template within the cycle header is used by stations on the ring to identify the data within each WBC as being either packet or isochronous. In the absence of errors only the cycle master may change the programming template . For open WBCs, packet data is delivered to the P-MAC, and isochronous data to the I-MAC. The I-MAC divides each isochronous WBC into individual transmission channels using the steering map. It delivers the data from the transmission channels it has open to the CS-MUX.

Both the cycle master and the slave stations monitor the other fields of the cycle header to ensure that correct ring synchronization is being maintained.

8.1.1 Ring initialization

In general the hybrid ring is initialized in basic mode via the MAC Claim process. Once the Claim process has been completed, bandwidth must be reserved for hybrid mode.

The station management (SMT) in a monitor station operating in basic mode, which wishes to take the ring into hybrid mode, shall ensure that the sum of the bandwidth allocated to synchronous packet traffic and the extra bandwidth required for hybrid operation does not exceed the allocatable synchronous bandwidth of the ring. The synchronous bandwidth which must be reserved by a monitor for hybrid operation is equal to the sum of the bandwidth for the cycle header, the bandwidth which is to be allocated to the isochronous WBCs, and any synchronous bandwidth required by the monitor station for mode switching or programming changes.

Mechanisms are provided within HRC for the monitor station taking the ring to hybrid mode to either assume the role of the cycle master, or to allow the monitor contention procedure to select the cycle master. The actual route taken in any particular instance will depend on prevailing policy on the ring.

The following subclauses describe ring initialization in general terms, a more detailed description of the ring synchronization process is given later.

8.1.1.1 Ring initialization by a pre-assigned cycle master station

This subclause will consider the case of a station wishing to become cycle master without going through the monitor contention procedure. The monitor contention procedure is described in a later clause.

Once synchronous bandwidth has been reserved for hybrid operation the monitor station wishing to become the cycle master may switch from basic to hybrid mode if the following conditions have been satisfied:

a) No basic mode only station on the ring.

b) No other cycle master on the ring.

c) The monitor station captures the token, or wins the Claim process.

While the token is captured, the monitor transmits a cycle onto the ring with the C1 (Synchronization Control) field set to R, the C2 (Sequence Control) fields set to S, CS set to a valid cycle sequence number, and the programming template as assigned by the local SMT entity. The monitor issues a new token and replaces the remaining Packet Data channel with fill symbols. If the ring is longer than 125 µs, the monitor continues to transmit new cycles until the first transmitted cycles return.

Upon receiving the starting delimiter symbol pair JK followed by 2 Control Indicator symbols (RR, SS, RS or SR) indicating the cycle control fields C1 and C2, other stations on the ring enter hybrid mode and propagate the hybrid cycle headers. Also since the cycles transmitted by the cycle master contain a sequence number, CS greater than 64, these stations are initialized to the slave mode of operation.

The monitor waits until it is sure that it is correctly receiving cycles it initiated, before it assumes the role of cycle master. The ring is now operational in hybrid mode. Transmit and receive data are connected through to the P-MAC and I-MAC according to the programming template. Cycles are transmitted with the C1 and C2 fields set to SS.

8.1.1.2 Ring initialization through monitor contention

The monitor contention procedure selects the highest ranking monitor on the ring to become cycle master. monitor stations may have pre-assigned ranks, or they may use a station management (SMT) protocol which makes use of the logical ranking scheme outlined in Annex F to establish their rank. Rank assignment may occur in either basic or hybrid mode. During the monitor contention procedure, all participating monitor stations generate cycles with C1 and C2 fields set to R and the cycle sequence field set to the station's monitor ranking. The station with the highest monitor ranking becomes the cycle master. If any station is only capable of basic FDDI operation, the ring returns to basic mode.

8.1.2 The cycle synchronization process

This clause provides an overview of Cycle Synchronization. In order to minimize complexity, the description within this clause concentrates on the flow of control during cycle synchronization and does not deal specifically with the flow of data through the H-MUX, which is described elsewhere.

Cycle synchronization is effected by use of the elements of the cycle header: the start delimiter (SD), the synchronization control symbol (C1), the sequence control symbol (C2), and the cycle sequence field (CS). The composition of these fields and their position in the cycle header is described in 7.2.

The precise synchronization process differs depending on the initial state and configuration of the ring. Two particular instances will be described: the first is synchronization of the ring by a pre-defined cycle master, the second is synchronization through the monitor contention procedure.

8.1.2.1 Ring synchronization by a pre-assigned cycle master

Once all the criteria for ring initialization described in 8.1.1.1 have been satisfied the SMT in this monitor may signal HRC_Start(master) to HRC. A transition to hybrid mode is then effected by HRC going directly into the RESYNCH state. It now adopts the role of a cycle master station.

The master transmits JK RS [CS], where $64 \leq CS \leq 255$ …

Any station, not of BASIC class, receiving JKCC (where C= R or S) in basic mode enters hybrid mode as a slave. Moreover a station receiving the first of these cycles knows, since CS >= 64, that CS is a sequence number and therefore infers that a cycle master is trying to synchronize the ring. It consequently resynchronizes to the received cycle, stays in slave mode, and sets CSexp equal to next(CSrc). However since CSrc is unlikely to be equal to the station's old CSexp value, the Latch_Sequence condition is false, so the slave sets C2tx = R.

Slave stations transmit JK RR [CS] …

This first cycle propagates round the ring in this form initializing all the stations on the ring to slave mode.

On the next cycle a slave station receiving JK RS [CS] … compares CSrc with CSexp and since, barring transmission errors, they are equal, the Latch_Sequence condition becomes true. The station now latches CSrc into CSopr and sets C2tx = S.

Slave stations transmit JK RS [CS] …

This second cycle propagates round the ring in this form, synchronizing all the stations on the ring and confirming their operation in slave mode.

The master continues to transmit cycles of the form JK RS [CS] … for a time T_Hold, which is the maximum time for a cycle to travel round the ring (default 13 cycles). It then checks received cycles for CCrc = RS, and for CSrc between the first transmitted CS (CSopr) and the last transmitted CS (CSreq). The master can not know the exact value of CS to expect since there is a variable delay of up to C_max cycles round the ring. Once two sequential cycles with correct headers have been received cycle control enters the MASTER state. While in this state the station sets C1tx=C2tx=S, to indicate that full synchronization has been achieved, and continues transmitting the cycle sequence in the CS field.

The master transmits JK SS [CS], where 64 ≤ CS ≤ 255 …

If the master fails to receive cycles correctly and failures persist for a further T_Recov (following T_Hold) the station may assume that it is unable to synchronize the ring (perhaps due to multiple stations in the RESYNCH state) and returns to basic mode.

On receiving CC = SS slave stations assume that the ring is fully synchronized and enter the fully operational steady state.

8.1.2.2 The monitor contention procedure

In the event of SMT issuing an HRC_start command with the contend parameter set or of a fault occurring in the cycle master, synchronization begins with the monitors contending to become cycle master via the monitor contention protocol. This protocol relies on the monitor ranks having been assigned by SMT. Normally the ranks would have been assigned prior to monitor contention taking place, however even if no ranks have been assigned it should be possible to negotiate ranks during the contention process itself. The mechanism for assigning monitor ranks operates in such a way as to avoid duplication of ranks, since if two stations on a ring have the same monitor rank the monitor contention procedure will not function properly. However even with duplicate rank assignment the probability of multiple master stations emerging is remote.

The monitor contention procedure involves all contending monitors going into the STANDBY state. In the STANDBY state cycles with C1=C2=R and CS = rank are sourced using local 8 kHz cycle timing.

Monitors source JK RR [rank] …

All SLAVE class stations on the ring resynchronize to the incoming cycles and simply repeat the cycle headers unchanged.

If the ring contains a monitor in the RESYNCH or MASTER states this monitor discards the contention cycles and continues to transmit its own cycles of the form JKRS[seq]. All stations receiving valid sequence numbers go into the SLAVE state and repeat this cycle header: this effectively ends monitor contention. If there is no such monitor on the ring, monitor contention continues as follows.

A monitor receiving cycles of the form JK RR [CS] … (where 0<CS<64) checks its contend flag and compares the received CS value with its own rank.

If the monitor is in the SLAVE state then

> If the contend flag is set and CS < monitor's own rank and, the transition would not cause data disruption due to LAB switching or the normal Latch_Rank conditions are satisfied or CSopr < 64 indicating that the previously latched CS value was a rank, then the monitor goes into the STANDBY state and commences transmitting its own contention cycles. Otherwise it remains in the SLAVE state.

If the monitor is in the STANDBY state then

> If CSrc > monitor's own rank and; the station has been in this state for at least T_hold and so has had a chance to fill the ring with its own rank or, the currently latched rank is null and the station is not currently scrubbing or, the normal Latch_Rank criteria are satisfied and the received rank is not equal to the currently latched rank; then the monitor backs off into the SLAVE state and starts repeating cycles.
>
> If CS = monitor's own (non zero) rank, and the same CS value has been received twice sequentially, and the station has been in the STANDBY state for at least T_Hold, then the monitor assumes that this is one of its own cycles which has gone round the ring. This implies that it is the highest ranking monitor and may therefore attempt to become the cycle master by going into the RESYNCH state.

Once a monitor is in RESYNCH state it sets C2=S and starts transmitting the cycle sequence in CS (a value between 64 and 255).

This is in effect the end of the monitor contention procedure. Ring synchronization now proceeds as described for the a pre-assigned cycle master.

8.1.2.3 The steady state

In the fully synchronized steady state the cycle master transmits cycles with the header JK SS [Seq]. The cycle timing used may either be derived from the station's local timing or from some external frequency reference such as ISDN. In the absence of errors slave stations repeat the cycle header unchanged. Data transfer takes place normally and the cycle master may change the programming template as required.

Both the cycle master and the slave stations continually monitor received data to ensure that correct ring synchronization is being maintained.

A master station receiving a cycle header resets TVS if CCrc = SS and the value of CSrc is equal to the expected value next(CSopr). A master continues transmitting cycles normally until consecutive errors have continued to occur for long enough for TVS to expire. Once this threshold of errors has been exceeded the cycle master assumes a serious error and goes back into the RESYNCH state to attempt resynchronisation. If this fails to recover synchronization the cycle master may go back to basic mode to allow another station to attempt to recover the ring.

Error recovery in a slave depends to a large extent on the use of the TVS and TNS timers along with their associated flags. These are used to ensure that cycles are received sequentially, and to facilitate recovery from error conditions when they are not received sequentially.

A slave station receiving correctly a cycle header with C2rc = S checks the value of CSrc against the expected value next(CSopr). If CSrc is equal to the expected value the cycle header is repeated as received, TVS and TNS are reset, and CSopr is loaded with the value of CSrc. However if CSrc is not equal to the expected value, the timers are not reset, CSopr is simply incremented (mod 192), and the cycle header is repeated with the C2 value changed to R to indicate an error in CS.

A slave station receiving correctly a cycle header with CCrc = SR repeats the cycle header as received. The actions taken are the same as those for an out of sequence CSrc value: CSopr is incremented (mod 192), and the TVS and TNS timers are not reset.

TNS expiry with the Hold flag reset, or TVS expiry with the Scrub flag reset, are, in general, indications that persistent cycle sequence or synchronization errors have been detected and that ring recovery is required. Exactly which one of these conditions triggers recovery depends on the detail of the error condition, the ring topology and the setting of ring parameters. monitor slaves detecting one of these conditions enter the STANDBY state and commence sourcing cycles with CCtx set to RR and CStx set to the station's monitor rank.

A monitor class station operating in slave mode which receives a cycle header with CCrc = RR checks its contend flag and checks the CSrc field for a rank value. If the conditions for entering monitor contention are not satisfied then the frame header is simply repeated as above. Otherwise the monitor discards the received cycle header, enters the STANDBY state and transmits its own cycle with CCtx=RR and CStx=Own rank.

A slave station receiving an early cycle header resynchronizes to the early cycle, and sets C1tx=R. The rest of the operation is the same as for receipt of a normal cycle.

If a cycle is late arriving at a slave station two actions are possible. If the station is a monitor and the sustain flag is set then it sets CCtx=RR and CStx=0 and generates a new cycle. If the station does not have the sustain flag set it sends idle symbols following the end of the transmitted cycle, until a new cycle header is received. If no cycles are received over a period of time, eventually TVS will expire at which point a monitor station goes into the STANDBY state and initiates monitor contention to try to recover ring synchronization If no monitors are present timer expiry eventually forces the ring back to basic mode.

8.1.3 Normal hybrid mode operation

8.1.3.1 Cycle master station

Cycles are transmitted in response to Cycle_Request signals generated from an 8 kHz external or internal reference. The cycle control fields C1 and C2 are set to S.

Normally, cycles return between the 8 kHz Cycle_Request signals. Each returning cycle is kept in the LAB until the next Cycle_Request signal is asserted. The Isochronous WBCs are delivered from the LAB to the I-MAC after the next HI_CYCLE_SYNC.indication. The Packet WBCs are delivered from the LAB to the P-MAC as soon as possible, without waiting for the next HI_CYCLE_SYNC.indication, to optimize the performance of the Timed token protocol.

8.1.3.2 Slave station

The slave station uses each cycle's programming template as input to the WBC template filter state machines. The state of each WBC state machine identifies the contents of the corresponding WBC to be either packet or isochronous data. Based on the states of the WBC state machines, the received WBCs are distributed to either the P-MAC or the I-MAC. Data received from the P-MAC and the I-MAC are merged and transmitted according to the states of the WBC state machines.

8.1.4 Changing the programming template

8.1.4.1 Cycle master station

The cycle master is responsible for changing the programming template based on the Assign WBC request from SMT. To modify the programming template, the cycle master SMT function shall capture the token via the P-MAC, or employ some alternative method to ensure that data in the packet channel is not lost due to the programming change.

Before the token is captured the cycle master shall ensure that it has enough bandwidth available to enable it to hold the token until the next cycle header, containing the new programming template, is transmitted. This bandwidth may be synchronous or asynchronous: for synchronous bandwidth to be used at least 125µs per token rotation must be available; for asynchronous bandwidth to be used the token must be at least 125µs early. After capturing the token, the cycle master then transmits a cycle with the new programming template. A new token is then issued immediately by the P-MAC.

> WARNING: If the programming template is changed when the token is not being held, the cycle master may be capable of buffering the effect of the change in bandwidth in the packet data channel in such a way as to ensure that no packet data is lost. The required buffering capacity is a function of the magnitude of the programming template change and the total latency of the ring. If a slave has its LAB engaged, the slave may also be capable of buffering the effect of the programming change; however, packet data could be lost due to packet channel underflow or overflow in the slave.

8.1.4.2 Slave station

Slave stations switch to the new programming template upon receiving the new cycle header from the cycle master. A slave station that has its latency adjustment buffer engaged shall not repeat any new SDU received in the packet channel after the latest programming change until the corresponding programming template has been repeated, to ensure that data in the packet channel is not discarded by the cycle master due to the programming change.

8.1.5 Returning to basic mode

An HRC entity shall always return to basic mode whenever its associated P-MAC entity enters the Claim or Beacon states, in order to allow these MAC frames to be transmitted in BASIC mode. Entry by the local P-MAC entity into one of these states shall have the effect of producing an HRC_Stop indication to the local HRC.

In addition, in order to allow ring recovery by MAC frames (Claim and Beacon) in basic mode, an HRC entity will return from hybrid to basic mode upon receiving the starting delimiter symbol pair JK followed by a valid frame control field of the form *MN*, where *M* = (8 or C) and *N* = (0 through F)) that identifies a MAC frame or token.

In general SMT in any station may return the ring to basic mode at any time, normally however a controlled return to basic mode is most efficiently carried out by the cycle master station.

Once in basic mode H-MUX asserts HI_INV to indicate to the I-MAC that the ring is no longer in hybrid mode. All symbols are delivered to the P-MAC. The hybrid mode signal is reset.

8.1.5.1 Cycle master station

SMT in a station wishing to return the ring to basic mode would normally first capture the token (to prevent data loss in the packet channel). SMT may then take a station into basic mode in a number of ways depending on the desired overall effect. A temporary return to basic mode can best be effected by SMT issuing an HRC_Stop primitive to the HRC and then transmitting basic mode MAC Claim frames. When the P-MAC in the (former) cycle master receives its own (or higher) Claim frames, the ring has been purged of cycles. The winning P-MAC issues a token in basic mode. This process takes the ring to basic mode but does not prevent subsequent return to hybrid mode. If a permanent return to basic mode is required SMT may effect this by setting its Class to Basic. This prevents subsequent return to hybrid mode.

8.1.5.2 Slave station

slave stations returns to basic mode upon receipt of a MAC frame header of the form JK*MN*, where *M* = (8 or C) and *N* = (0 through F).

8.2 Error recovery

8.2.1 Error conditions

This clause considers the following error conditions and the associated recovery procedures:

a) Corrupted cycle header
b) Late token in Packet Data Channel
c) Loss of Synchronization
d) Failure to Synchronize

8.2.2 Corrupted cycle header

8.2.2.1 Starting delimiter

Transmission errors can cause the physical layer to lose the JK symbol sequence at the beginning of a cycle or to create a JK symbol pair within a cycle. To avoid loss of synchronization due to false JKs, cycle holding mode is introduced to the PHY.

In the cycle holding mode, a new starting delimiter is not accepted during the cycle. If a new starting delimiter is recognized, it is reported to H-MUX as Violation symbol(s), and the symbol boundary is not changed; however, cycle holding mode is cleared. A second starting delimiter received within the same cycle is reported as a starting delimiter, since the probability gives the frequency of occurrence of a second false JK in the same cycle as less than one in 100 000 years.

When not in the cycle holding mode, any starting delimiter is indicated to H-MUX as a starting delimiter, rather than as Violation symbol(s).

8.2.2.2 Corrupted cycle control symbols

Corrupted values of C1 and C2 are translated to R symbols and are then treated as such.

8.2.2.3 Corrupted cycle sequence values

Cycles are numbered by the cycle master from 64 to 255. When a cycle is received, all H-MUX entities compare the current received cycle sequence value with an expected cycle sequence value. During normal operation the expected value is set equal to the last received value plus 1 (mod 192). If the received and expected values are not equal, as for the receipt of a corrupted CS value, the received CS value is flagged as being in error. Slaves repeat these CS values as received however the C2 value in the corresponding cycle is changed from S to R to indicate that cycle sequence Error has been detected.

Persistent cycle sequence errors are checked by using the TVS and TNS timers. On a timer expiry indicating persistent cycle sequence errors a monitor normally infers that the cycle master is not functioning correctly and so begins to source cycles with the C1 and C2 fields set to R and the cycle sequence value set equal to the station's monitor ranking. This initiates the monitor contention procedure where the participating monitor with the highest rank becomes cycle master. All WBCs remain active, but no re-programming is possible until a cycle master takes control.

8.2.2.4 Programming template

If the programming template symbol is not decoded as R or S, the H-MUX replaces the corrupted value with either an S or a T symbol, depending on the state of the corresponding WBC state machine. If the WBC was previously assigned as isochronous the corrupted value is replaced with an S symbol to maintain continuity of isochronous service. If the WBC was previously assigned as packet, or if no previous assignment is known, the corrupted value is replaced with a T symbol to avoid undetected data corruption in the packet channel.

slave stations repeat the altered programming template and use it to sort the WBC. If an altered symbol corresponds to an Isochronous WBC, H-MUX continues to provide isochronous service on the WBC. If an altered symbol corresponds to a Packet WBC, H-MUX repeats the WBC while transmitting HP_INV to P-MAC.

The cycle master restores the correct values by transmitting the correct programming template in every cycle. Consequently, the altered programming information should be corrected on the next cycle.

The physical encoding of R, S and T requires multiple errors on the same link to change one of these three symbols into another one.

NOTE 2 The probability gives the frequency of occurrence of this error in a programming template as less than once in 10 years on a very noisy link, and as less than once in millions of years under normal circumstances. Depending upon the frequency of programming changes, the template filtering function in a slave station could cause a corrupted R to become an S less than once in a few weeks to a few years on a very noisy link, and less than once in thousands of years under normal circumstances. This error would result in corruption of the packet channel for one cycle that is undetected by H-MUX and unreported to P-MAC. Given a 32 bit P-MAC frame Check Sequence, the probability of undetected packet corruption due to such errors is less than 10^{-20} under extremely pessimistic assumptions, and much better in normal operation.

8.2.3 Late token in the packet data channel

There are two steps in recovering the late token in an FDDI-II ring: Purge and Claim processes.

The Purge frames are generated by the cycle master to quickly re-establish the P-MACs. If the Purge frames do not solve the problem, all stations return to basic mode and enter the Claim Process.

8.2.3.1 Purge process

TVX expiration generally indicates that the token has been lost. TRT expiration generally indicates that one or more stations do not have consistent P-MAC scheduling parameters (e.g. Target token Rotation Time (TTRT)). In order to quickly re-establish the ring, all stations must use the same token Rotation Timer that has been agreed upon during the Initialization process.

The P-MAC in the cycle master generates Purge frames that contain the TTRT value, without leaving hybrid mode. Upon receiving the Purge frame, P-MACs copy the TTRT value and repeat the frame. No frames can be transmitted by the P-MACs until the P-MAC in the cycle master receives its Purge frames and issues a new token.

The Isochronous WBCs are not affected by the Purge process.

8.2.3.2 Claim process

After the Purge process, all stations are supposed to operate with the newly recorded TTRT value. If the token continues to arrive late, stations return to basic mode to enter the Claim process.

When the Claim process has completed successfully, a new token is generated. The cycle master captures the token and transmits a new cycle. The ring re-enters hybrid mode.

The Isochronous WBCs are disabled while stations return to basic mode.

8.2.4 Loss of synchronization

A cycle is expected every 125 µs, and its arrival is indicated by H-MUX with the HI_CYCLE_SYNC.indication signal.

The cycle is transmitted by the cycle master based on an external clock if available. If the external source is not available or fails to provide the HM_CYCLE_SYNC.request primitive, the cycle master generates cycles with its own clock.

If the start of a new cycle is recognized by H-MUX before the previous cycle completes, the Early_Cycle Counter is incremented and C1 set to R.

If the start of a new cycle is not received within a specified time after the end of the previous cycle, the cycle master may not be functioning properly. monitors which fail to receive a cycle within the permitted window may generate a new cycle with CCtx=RR, providing the sustain parameter is set. If no cycles are received for a period of time, a time-out indicated by TVS expiry with the Scrub flag clear will occur. At this point the station goes into the STANDBY state where it initiates the monitor contention procedure by sourcing cycles with C1 and C2 fields set to R and the cycle sequence field set to the station's monitor ranking. If there is no cycle master on the ring the participating monitor with the highest rank becomes cycle master.

8.2.5 Failure to synchronize

On transition to hybrid mode HRC maintains a time-out on each stage of the synchronization process. Failure to complete that stage of synchronization within this time-out results in stations returning to basic mode. At this point management procedures can be used to ascertain why hybrid synchronization was not possible. In addition to this, SMT may maintain a time-out on the initial synchronization process and all subsequent resynchronisation processes. Failure to achieve full synchronization, with CC = SS, within this time-out can result in SMT taking the ring back to basic mode and initiating recovery via the Claim process. Once the Claim process has been completed initialization to hybrid mode may be re-attempted.

8.3 Structure

In order to reduce the complexity of the description of the hybrid multiplexer (H-MUX) process it is advantageous to distribute the overall scope of all functions over various subprocesses. Moreover a need for separate processes arises from the requirement that certain functions in one process need to be performed concurrently with and asynchronously to the states of other processes. In the case where identical functions are needed by two or more processes they have either been placed in one process with the appropriate signals sent to the other processes or they have been duplicated. In either case signals may be replaced with their corresponding functions or vice versa. It should be stressed that the aim of this partitioning is to produce a well defined reference model for the behaviour of H-MUX and not to imply any particular implementation. As a consequence some functionality may have been included in the description of a particular process that may be more efficiently implemented as a part of some other entity. If this is the case, or for any other reason, any implementation is free to change the order of the processes shown or to redistribute some, or all, of the functionality, of any one or all of the processes, into some other entity(s), so long as complete interoperability with the reference model of HRC described in this document is maintained.

When dividing up the process the following points were taken into consideration:

- Grouping of similar functions, including logically related interface elements
- Definition of process states which should be independent from the states of other processes where possible
- An external event (except perhaps reset) should affect only one subprocess.
- There should be a minimum number of additional interface elements for internal process communication.

A distribution of functions which takes most of these requirements into consideration is shown in figure 13. H-MUX is described in terms of four cooperating asynchronous processes: cycle acquisition, cycle generation, Latency Adjustment, and cycle exchange. These processes operate independently, their operation being synchronized by signals (e.g. RX_Start, TX_Start) and exported variables .

The H-MUX processes are defined as cooperating state machines. It is assumed that time elapses only within discrete states, and that state transitions are logically instantaneous. It follows that actions requiring more than one symbol/byte time to complete shall be performed within states; however when these actions are associated with a specific transition, they may be described as part of that transition. In all such cases the actions occur prior to the associated transition. Actions described as part of a state occur each time the state is entered. Thus when a triggering event occurs externally to a state machine, the state machine shall perform the following event processing sequence as a logically atomic operation:

a) Evaluate all conditions within the current state.

b) If the conditions for a state transition are satisfied, then

  i) perform the transition actions in the current state;

  ii) enter the new state;

  iii) perform the entry actions (if any) for the new state;

  iv) if an immediate transition from the new state is possible, then repeat the sequence beginning with step 1.

c) If the conditions for in-state actions are satisfied, then the specified actions shall be performed.

Event propagation (signals) between processes may incur some conceptual propagation delay. A signal and its parameters shall always experience the same identical delays between processes so they arrive at the same conceptual time and in the order they were generated. The actual values of these delays are, to a large extent, implementation dependent and are not specified in this document unless they affect interoperability.

The H-MUX state machines are specified both with text and with state diagrams. In the state diagrams, states are shown as vertical bars and state transitions as horizontal arrows, with the triggering event or condition above the arrow and any action below the arrow. In the event of any discrepancy, the state diagrams and attached notes take precedence over the text.

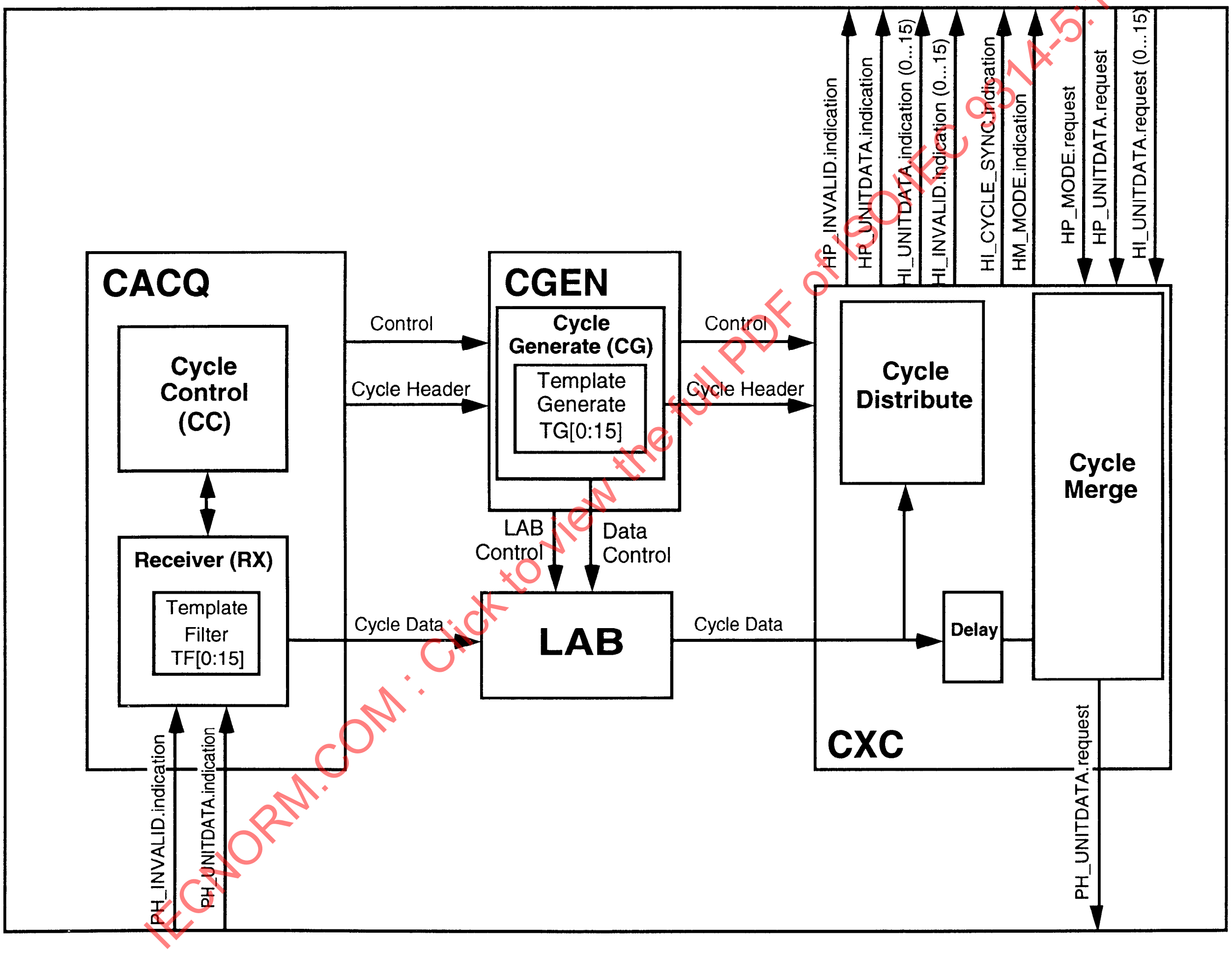


Figure 13 — H-MUX structure

8.3.1 The hybrid multiplexer processes

8.3.1.1 Cycle acquisition (CACQ)

The cycle acquisition process consists of two sub-processes: Receive Control and cycle control.

The Receive sub-process continually monitors data received at the PHY interface through PH_UNITDATA.indication primitives. It is responsible for identifying the start of received cycles, the termination of cycles, and the receipt of MAC frames. The occurrence of these events is communicated to other processes within H-MUX via the RX_Start, RX_Abort, and RX_MAC primitives respectively.

In addition, the Receive sub-process breaks the incoming cycles down into cycle header and data fields. The cycle header field is used by the cycle control sub-process, while the data field is sent to the LAB (or CXC in slave only stations). Since the packet and isochronous WBCs may be treated separately by the LAB, the Receive sub-process may be required to separate the data according to the template. The operation of the Receive sub-process on receipt of a cycle is described by figure 14, the HRC Receive State Machine and figure 15, the WBC template filter State Machine.

The cycle control sub-process works in conjunction with the Receive sub-process to continually monitor received data for valid cycles and MAC frames. The C1, C2, and CS fields following received start delimiters are checked, processed and the appropriate actions taken. The operation of the cycle control sub-process is described by figure 16, the HRC cycle control State Machine.

8.3.1.2 Cycle generation (CGEN)

The cycle generate process maintains control over the cycle transmission process to ensure that data transmitted onto the ring is consistent with the synchronization state of the hybrid ring. The cycle generation process is only required in monitor stations, where it maintains control over the cycle transmission process to ensure that data transmitted onto the ring is consistent with the synchronization state of the hybrid ring. To this end, the cycle generation process generates the C1, C2, CS and programming template (PT) fields of the transmitted cycles based on its current state and the control and header information from cycle acquisition. Cycle acquisition passes the required CC, CS and PT fields to the cycle generation process. The transmitted PT field is constructed by following figure 18, the HRC WBC template generation State Machine.

In monitor stations, the control and header information from cycle acquisition is used to load the cycle data into and out of the LAB and will have to be kept in step with this data while it is in the LAB.

The header fields from the cycle generation process are passed to the cycle exchange process (CXC). Cycle generation signals TX_Start whenever a new cycle is to be initiated.

The cycle generation process also generates the 8 kHz timing used for transmitting cycles. In master mode this timing may be based on a local frequency reference or it may be synchronized to an external 8 kHz timing reference provided by the SM_HM_CYCLE_SYNC.request primitive. In slave mode this timing information is derived from the received cycles via the RX_Start primitive.

8.3.1.3 Latency adjustment buffer (LAB)

The latency adjustment buffer is required in monitor stations but need only be engaged while operating as the cycle master. The LAB is permitted to be engaged while operating as a slave; however, this option should be exercised carefully, since it can produce excessive ring delays.

The primary use of the LAB is to ensure that isochronous data takes an integral multiple of 125 µs to travel around the ring.

Since the dedicated packet group and any WBCs assigned to packet data are pooled into a single packet channel, packet data should be allowed to bypass any isochronous data in the LAB, assuming the first available position on the outgoing side. Also, in the cycle master or when operating as a slave station with the LAB engaged, the transmitted template may be different from the received template. Thus, data must be loaded into the LAB under the control of the cycle acquisition process, and taken out of the LAB under the control of the cycle generator process.

A slave station that has its latency adjustment buffer engaged shall not repeat any new SDU received after the latest programming change until the corresponding programming template has been repeated. During both master and (optional) slave modes of LAB operation, if a programming change occurs within a SDU received in the packet channel, that SDU should be repeated unless it causes the packet channel to underflow or overflow. If a SDU received in the packet channel is partially repeated, that SDU shall be explicitly aborted by generating a LL or LI symbol pair.

> WARNING: During both master and (optional) slave modes of operation, the latency adjustment buffer should be designed to avoid losing the token in the packet channel during programming changes or any other events that may cause buffer overflow or underflow.

The LAB and its associated control logic (which may reside in the CACQ and CGEN processes) should be able to deal with or avoid the following possibilities:

Overflow: The LAB should have sufficient capacity for 1 cycle plus any expected jitter plus any expected ring length fluctuations. The cycle is, of course, a defined length; the jitter should be down to two symbols and the effective physical length of the ring should not fluctuate by more than a few symbols due to temperature, etc without implying a fault or a ring configuration change. Although the required LAB capacity can vary, a capacity of greater than one (1) cycle is required.

Underflow: If a ring becomes operational with a length of slightly less than an integral number of cycles, the LAB in the master station will be nearly empty. If the effective length of the ring then increases to slightly more than this integral number of cycles, CGEN should be able to generate a cycle to maintain the ring while the expected cycle arrives and is loaded into the LAB. This avoids an underflow of the LAB and results in the LAB containing almost a full cycle. If then the effective physical length of the ring decreases, the LAB will be required to expand further. The LAB will then contain slightly more than one (1) cycle but will be immune to further underflow.

The definition of CGEN specifies the initial contents of supplied cycles.

Pre-emption: If the control signals required to drive CGEN are passed to it as they are generated but with the same time displacement as the data through the LAB, then it is possible that the CGEN process may be triggered by the RX_Start signal which came from the incoming starting delimiter only to be halted by a return to basic mode as a result of the following bytes. In this case, CGEN will generate a cycle fragment of an implementation dependent length which may upset downstream HRCs.

It is suggested that the CC-State information be established before the RX_Start signal is passed on from the Receive process.

Further, the LAB is reset by certain cycle control state transitions and on certain error conditions. The LAB shall be reset by the HRC_Reset signal on entering basic mode. Other causes for LAB reset are left to the implementor.

In the absence of the LAB in a slave station, data passes directly from cycle acquisition to cycle exchange.

8.3.1.4 Cycle exchange (CXC)

The primary function of the cycle exchange process (CXC) is to route any WBCs open at this station out of the stream of cycle data to the P-MAC and I-MAC entities, insert data from the P-MAC and I-MAC back into these WBCs and combine the new cycle data with the header information from the cycle acquisition or cycle generation process to form the cycle to be transmitted to PHY.

The HM_MODE.indication and HI_CYCLE_SYNC.indication primitives are generated by the cycle exchange process. The HM_CYCLE_MODE.indication primitive provides an indication of the current mode of operation to both I-MAC and P-MAC, while the HI_CYCLE_SYNC.indication provides an 8 kHz phase reference to I-MAC so that data transfer to and from the I-MAC is correctly synchronized with the current cycle.

The cycle distribute sub-process of cycle exchange accomplishes the transfer of data from the outgoing cycle to the P-MAC and the I-MAC.

The cycle merge sub-process combines the unchanged, repeated data with the new data from the P-MAC and the I-MAC and the cycle headers from the cycle generation process (or the cycle control process in the case of a slave-only station) into complete cycles which are transmitted to PHY with the PH_UNITDATA.request primitive.

Depending on the implementation, it may be necessary to insert a small delay between the cycle distribute and cycle merge sub-processes as shown in figure 13 to match the internal repeat delay with the delays between data indication and data request in the P-MAC and I-MAC. This delay should be kept to a minimum as it adds to the total ring latency.

8.3.1.5 Wideband channel control

The wideband channel control process is the part of H-MUX responsible for controlling access to and from the wide band channels within the cyclic groups. This function may be seen as consisting of two elements: WBC template control and HI-SAP template control. The first of these is concerned with determining the assignment of each WBC to either packet or isochronous data. The second identifies which HI-SAPS, corresponding to WBCs allocated to isochronous data, are open at this station: it is performed by SMT directly programming the template of open HI-SAPs (as Open[ISAP]) using the SM_HM_WBC.request primitive. SMT can also specify whether a P-MAC is available at the HP-SAP as Open[PSAP].

Due to the nature of the pipeline through H-MUX with different actions needing to be performed on the data and template at different points in the pipe, it is necessary to distribute the WBC template control function between the parts of H-MUX described previously. In general WBC template control consists of three separate components: the WBC template filter state machine in cycle acquisition, the WBC template generation state machine in cycle generation, and the WBC processing function in cycle exchange. The WBC template generation state machine is only required in a monitor.

WBC template control is described in terms of state machines. For each WBC state machine, each WBC has an associated state variable which determines how data in that WBC is treated. Therefore, for each of the WBC state machines described, there are conceptually 16 identical but independent state machines acting on a set of 16 state variables, corresponding to the 16 WBCs.

8.4 Cycle acquisition

The cycle acquisition process is described in terms of its sub-processes: receive control, WBC template filtering, and cycle control.

8.4.1 Receive control process

The receive process continually monitors data received at the PHY interface through PH_UNITDATA.indication primitives. It is responsible for identifying the start of received cycles, the receipt of MAC frames, and cycle synchronization errors. The occurrence of these events is communicated to other processes via the RX_Start, RX_CS and RX_MAC signals and the Cycle_error flag. In addition the receive process contains the WBC template filter functions associated with processing the received programming template.

The rest of this subclause specifies the states and state transitions of the receive control subprocess.

8.4.1.1 State RX0: SCAN

When the Receive Control process is initialized by the HRC_Reset signal it enters the SCAN state. This is the normal quiescent state of the Receive Control process. In this state the Receiver looks for a valid start of frame or start of cycle delimiter (JK) in the PH_UNITDATA.indication stream. On detection of a JK start delimiter the receive process issues an RX_Start signal.

RX(00a) The HRC reset and PH_Invalid signals or a change of Class to Basic while operating in hybrid mode, causes the state machine to be re-initialized.

RX(00b) If the start of a new cycle has not been received PA_Max symbols after the end of the last cycle (as indicated by entry into the RX0 state) a cycle error is flagged and counted.

RX(01) On receipt of a valid start delimiter (JK) a transition to state RX(1) CHECK_CC occurs and RX_Start is signalled indicating the possible start of a new cycle.

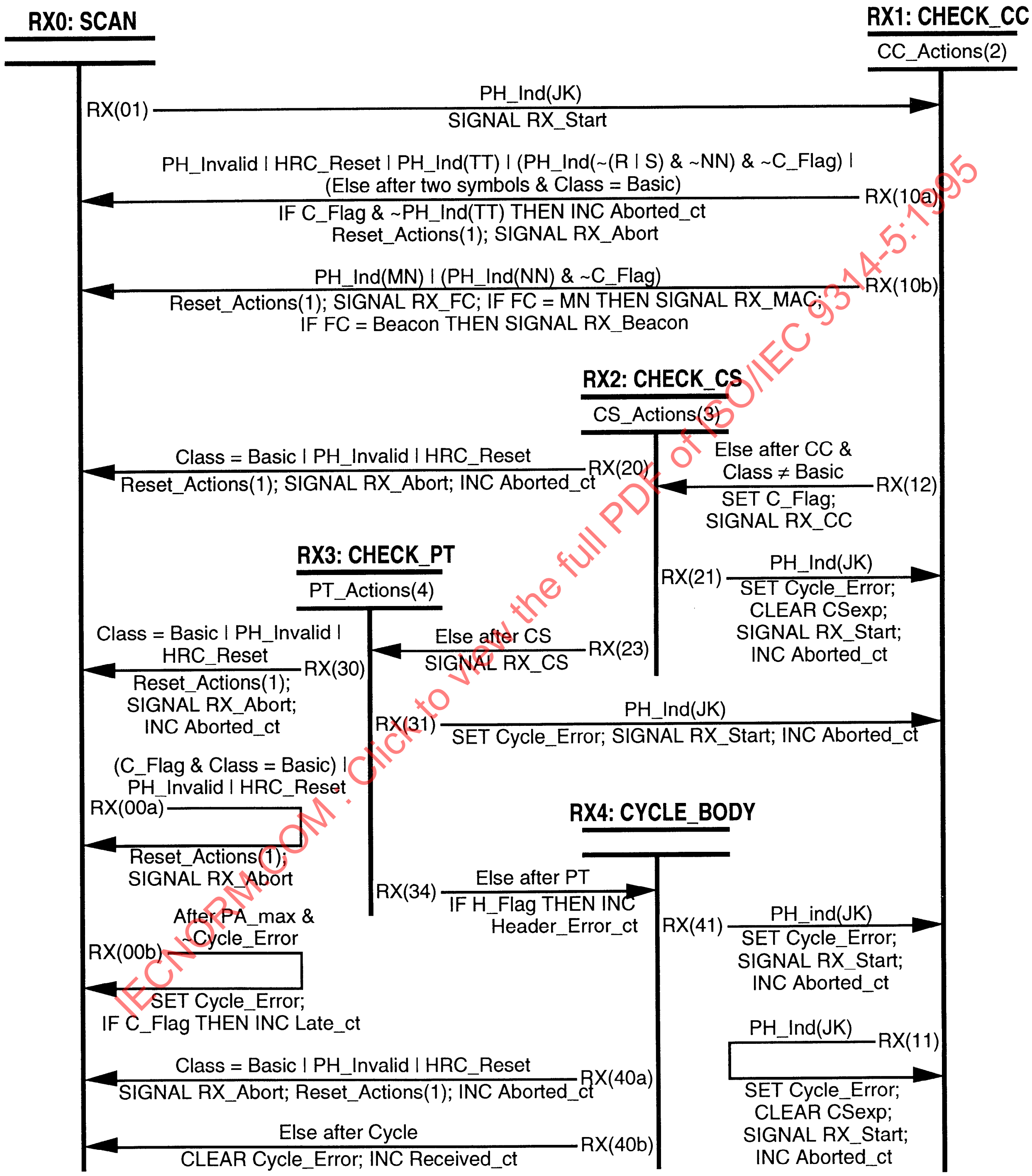


Figure 14 — HRC receive state diagram (Part 1 of 2)

HRC Receive Actions:

(1) Reset_Actions:

```
SET Cycle_Error;
CLEAR C_Flag, CSexp
```

(2) CC_Actions:

```
IF C1rc ≠ (R | S)
    THEN SET C1rc = T
IF C2rc ≠ (R | S)
    THEN SET C2rc = T
IF C1rc ≠ T & C2rc ≠ T
    THEN SET CCvalid; CLEAR H_Flag
    ELSE CLEAR CCvalid; SET H_Flag
```

(3) CS_Actions:

```
IF (CCrc = RR & CSrc = (nn)) | CSrc ≥ 64
    THEN IF CSrc = CSexp &
        ((CCrc = RR & 0 < CSrc < 64) |      {valid rank}
        (CCrc = (RS | SS) & CSrc ≥ 64))     {valid sequence}
            THEN SET CSvalid
            ELSE CLEAR CSvalid
        SET CSexp = next(CSrc)
    ELSE CLEAR CSrc, CSexp, CSvalid; SET H_Flag       {CSrc is illegal}
```

(4) PT_Actions:

```
FOR EACH PT[WBC] symbol DO
    SET PTrc[WBC] = Output(TF(PH_Ind)[WBC]);
    IF ¬PH_Ind(R | S | T)
        THEN SET H_Flag
```

Figure 14 — HRC receive state diagram (Part 2 of 2)

8.4.1.2 State RX1: CHECK_CC

In this state the Receiver scans the next symbol pair received on the PH_UNITDATA.indication stream, following the initial JK, for either a valid frame control field indicating a MAC frame or a valid cycle control field consisting of R and S symbols. The appropriate exit is taken according to the value of this symbol pair and the current mode of operation (frame or Cycle) as defined by the C_flag. The CC_valid flag is set if CCrc is a combination of R and S symbols.

RX(10a) The HRC reset or PH_Invalid signals, a current Class of Basic, receipt of an Abort sequence (JKTT) or receipt of two symbols which cannot be accepted either as a frame or a cycle control field, all cause the state machine to return to the SCAN state. RX_Abort is signalled to indicate the absence of a cycle and cycle error is flagged. Furthermore in frame Mode (C_flag reset) a control field that is neither a valid MAC frame control field nor a valid cycle control field is not processed any further so the state machine may be re-initialized back to SCAN. In hybrid mode, the Aborted_ct is incremented unless an Abort sequence was received.

RX(10b) On receipt of a frame control field indicating a basic mode frame or MAC frame the occurrence of the frame control field is signalled to cycle control via the RX_FC signal. If the frame control field indicates a MAC frame, then RX_MAC is signalled; if the FC field was that of a beacon frame then RX_Beacon is signalled also. Then the Receive state machine returns to SCAN.

RX(11) On receipt of a valid start delimiter (JK) a transition back to state RX1 CHECK_CC occurs. Cycle error is flagged to indicate the pre-emptive termination of a cycle, and the Aborted_ct is incremented.

RX(12) On receipt of a valid cycle control field indicating the arrival of a cycle, a transition to state RX2 CHECK_CS occurs. Furthermore in hybrid mode (C_flag set) a control field that is neither a MAC frame control field nor a valid cycle control field is interpreted as a corrupted cycle control field which is passed on to cycle control for further processing, therefore the possible error is ignored at this stage and a transition to state RX2 CHECK_CS is made. The RX_CC signal is sent to cycle control to indicate that it may now process the received cycle control field.

8.4.1.3 State RX2: CHECK_CS

In this state the Receiver, in the absence of a JK start delimiter, interprets the next symbol pair received on the PH_UNITDATA.indication stream as the cycle sequence byte CS. The CS_valid flag is set if the CS value is equal to the expected value and is consistent with the received CC field. The next expected value of CS received, CSexp, is derived from the received CC and CS fields.

RX(20) The HRC reset signal or a change of Class to Basic or receipt of PH_INVALID.indication, causes the state machine to be re-initialized to the SCAN state, RX_Abort is signalled to indicate the end of a cycle and cycle error is flagged, and the Aborted_ct is incremented.

RX(21) On receipt of a start delimiter (JK) a transition back to state RX1 CHECK_CC occurs. Cycle error is flagged to indicate the pre-emptive termination of a cycle, and the Aborted_ct is incremented.

RX(23) If the next received symbol pair is not a start delimiter it is interpreted as the cycle sequence byte CS. The Receive process now signals RX_CS to cycle control to indicate that it may now process the received cycle sequence field, and makes a transition to state RX3:CHECK_PT where the WBC template filter state machine acts on the received programming template.

8.4.1.4 State RX3: CHECK_PT

This is the normal state of the Receive control process during the reception of the programming template. During this state, the WBC template filter state machine is run for each WBC in the programming template.

RX(30) The HRC reset signal or a change of Class to Basic, or receipt of PH_INVALID.indication, causes the state machine to be re-initialized to the SCAN state, RX_Abort is signalled to indicate the end of a cycle and cycle error is flagged, and the Aborted_ct is incremented.

RX(31) Receipt of a valid start delimiter (JK) is interpreted as a possible start of the next cycle. A transition is therefore made back to state RX1 CHECK_CC. Cycle error is flagged to indicate the pre-emptive termination of a cycle, and the Aborted_ct is incremented.

RX(34) At the end of the PT, providing the next received symbol pair is not a start delimiter, a transition is made to state RX4 CYCLE_BODY where the Receive state machine stays during the body of the cycle. If a header error has been detected (H_Flag set) the Header_Error_ct is incremented.

8.4.1.5 State RX4: CYCLE_BODY

This is the normal state of the Receive control process during the reception of the body of a cycle. The incoming data is scanned for JK start delimiters which might indicate the start of the next cycle or frame.

RX(40a) The HRC reset signal or a change of Class to Basic, or receipt of PH_INVALID.indication, causes the state machine to be re-initialized to the SCAN state, RX_Abort is signalled to indicate the end of a cycle and cycle error is flagged., and the Aborted_ct is incremented.

RX(40b) At the end of a cycle (96 CDGs) the cycle error flag is reset and the Receive state machine goes to the RX0 SCAN state to wait for the start of the next cycle. The Received_ct is incremented for the completed cycle.

RX(41) Receipt of a valid start delimiter (JK) is interpreted as a possible start of the next cycle, a transition is therefore made back to state RX1 CHECK_CC. Cycle error is flagged to indicate the pre-emptive termination of a cycle, and the Aborted_ct is incremented.

8.4.2 WBC template filter process state descriptions

This subclause specifies the states and state transitions of the WBC template filter subprocess as shown in figure 15.

The major triggering event to each of the WBC template filter state machines is the receipt of the corresponding element (symbol) of the programming template within a hybrid cycle header. The elements of the received programming template identify the contents of the corresponding received WBCs to be either packet or isochronous data. Each cycle's programming template is copied into the WBC template filter, for use as input to the state machines. All symbols not recognized as R or S are translated to a T. In a non-monitor station the states of the WBC template filter state machines are used to generate the transmitted programming template in such a way that a slave station always repeats the received programming template with any symbols not recognized as R or S being translated to either an S or a T symbol. All stations monitor the received programming template and inform SMT of any error conditions.

8.4.2.1 State TF0: TK_INITIALIZE

When the WBC template filter process is initialized by the HRC_Reset signal it enters the TK_INITIALIZE state. During normal hybrid operation this state is entered only as a result of an error in the received programming template.

TF(00) PT symbol error: receipt of an illegal symbol for this WBC in the programming template indicates that the element corresponding to this WBC in the programming template has been corrupted. Since no positive indication about the contents of this WBC has been received for this cycle the state machine stays in the TK_INITIALIZE state, a T symbol being substituted in this field of the output template.

TF(01) Enable Packet Mode: receipt of an R symbol for this WBC in the programming template enables it for operation in Packet Mode. The R symbol is repeated in this field of the output template.

TF(02) Enable Isochronous Mode: receipt of an S symbol for this WBC in the programming template enables it for operation in Isochronous Mode. The S symbol is repeated in this field of the output template.

8.4.2.2 State TF1: TK_OPERATE

This is the normal operating state for a WBC template filter state machine corresponding to a WBC operating on packet mode while R symbols are being received in the corresponding symbol of the programming template.

TF(10a) PT symbol error: receipt of an illegal symbol for this WBC in the programming template indicates that the element corresponding to this WBC in the programming template has been corrupted. Since no positive indication about the contents of this WBC has been received for this cycle the state machine enters the TK_INITIALIZE state, a T symbol being substituted in this field of the output template.

TF(10b) HRC_Reset: receipt of a Reset signal causes the state machine to enter the TK_INITIALIZE state.

TF(11) Packet Mode: receipt of an R symbol for this WBC in the programming template maintains its operation in Packet Mode. The R symbol is repeated in this field of the output template.

TF(12) Enable Isochronous Mode: receipt of an S symbol for this WBC in the programming template enables it for operation in Isochronous Mode. The S symbol is repeated in this field of the output template.

8.4.2.3 State TF2: ISO_OPERATE

This is the normal operating state for a WBC template filter state machine corresponding to a WBC operating in isochronous mode. In this state received data in the corresponding WBC is treated as isochronous until either an R symbol is received in the corresponding symbol of the programming template or HRC_Reset is issued.

TF(20) HRC_Reset: receipt of a Reset signal causes the state machine to enter the TK_INITIALIZE state.

TF(21) Enable Packet Mode: receipt of an R symbol for this WBC in the programming template enables it for operation in Packet Mode. The R symbol is repeated in this field of the output template.

TF(22) Isochronous Mode: receipt of any symbol except R, for this WBC, in the programming template maintains its operation in Isochronous Mode. The S symbol is placed in this field of the output template.

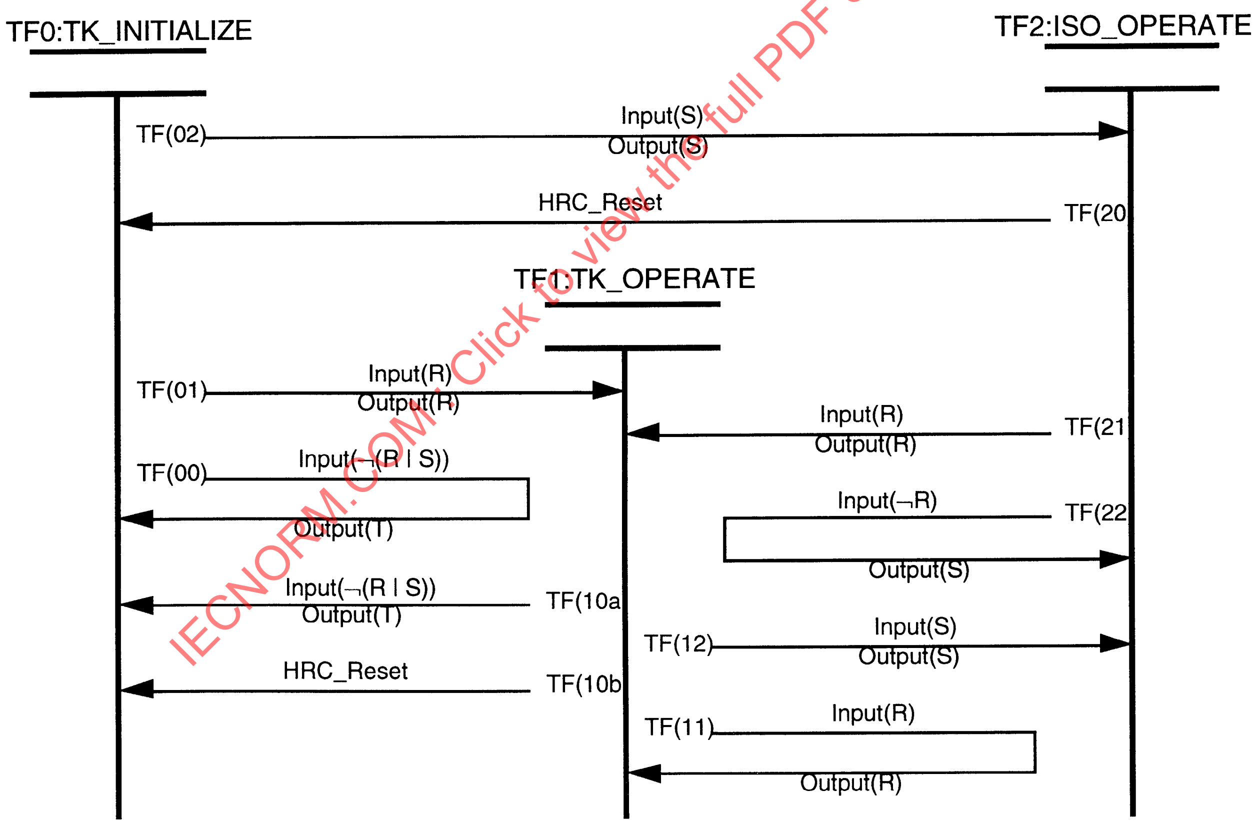


Figure 15 — WBC template filter state diagram

8.4.3 Cycle control

This subclause specifies the operation of the cycle control process as shown in figure 16. This is the main controlling process within H-MUX and is responsible for maintaining the synchronization state of the HRC entity.

8.4.3.1 State CC0: BASIC (basic mode)

The BASIC state is the initial state of the cycle control state machine, it is also the only state allowed in a BASIC class station. Cycle control being in this state results in basic FDDI operation, CXC routes frames directly between P-MAC and PHY: cycles are not processed. In this state rank may be (re)negotiated using SMT frames: any global reconfiguration may require a renegotiation of rank. Received data is monitored for hybrid cycle headers.

CC(00a) MAC frame Received: Receipt of a basic mode MAC frame (Beacon, Claim or Purge) causes a re-entry to the BASIC state. The main effect of this is to reset TNS and thus prevent a station with the Initialize flag set going into hybrid mode before ring initialization is complete.

CC(00b) Reset: The HRC_Reset signal causes the state machine to be initialized to the BASIC state, all monitor stations have their Class reset to slave.

CC(01) Cycle Received: A cycle header being received in the BASIC state implies that a station on the ring is attempting to initiate a transition to hybrid mode. Any SLAVE or MONITOR class station receiving a cycle header therefore effects an immediate transition to hybrid mode. Initially both types of station enter the SLAVE state to await the CS field. monitor stations can subsequently go to the STANDBY state if Recovery_Required conditions are satisfied.

CC(02) Initialization_Required: In basic mode a monitor continues to scrutinize data received from the ring for valid cycle headers or MAC frames. If the Initialize flag is set then, on expiry of TNS (or if TNS has already expired when the Initialize flag is set), a transition to hybrid mode occurs. Since it is possible for several stations to make this transition simultaneously the transition must be to the STANDBY state where the monitor contention procedure is initiated. A monitor in basic mode may also go into the STANDBY state by SMT signalling HRC_Start with the contend parameter set. This would normally occur on receipt of a token or if the token claim process was won by the station receiving its own claim frames. The Scrub flag is set and TVS is reset to T_Hold, on this transition so that old claim frames, or claim frame fragments may be scrubbed from the ring and do not interfere with hybrid initialization.

CC(03) Initialize_master: A monitor in basic mode may be made to go directly into the RESYNCH state by SMT signalling HRC_Start with the master parameter set. This would normally occur if the token claim process was won by the station receiving its own claim frames. This is a safe procedure since all the other stations are repeating claim frames at this time. This is the fastest way of initializing the ring into hybrid mode. The Scrub flag is set and TVS reset to T_Hold, on this transition so that old claim frames, or claim frame fragments may be scrubbed from the ring and do not interfere with hybrid initialization.

8.4.3.2 State CC1: SLAVE (hybrid mode)

In the SLAVE state a station synchronizes its transmit cycle timing to received cycles. The cycle synchronization, C1, and sequence, C2, fields of received cycles may either be reset or repeated, according to the synchronization status, however they may not be set. Both packet and isochronous data transfer can take place in this state however the quality of service may be degraded during synchronization.

While in this state Slave_Actions are carried out. These are primarily concerned with ensuring that synchronized cycles are received sequentially, and recovering from error conditions when they are not. Error recovery is based on the use of the TVS and TNS timers along with their associated flags.

TNS is used in conjunction with the Hold and Valid flags to control the latching of sequence numbers and ensure that correct sequences are maintained. Whenever a new sequence number is latched which is not equal to the next CSopr, TNS is reset to T_Hold, the Hold flag is set, and the Valid flag cleared: a holding state is entered where CSopr latching is inhibited until TNS expires. On TNS expiry the ring should be clear of stale cycles so the Hold flag is cleared and CSopr latching is re-enabled, TNS is reset to T_Valid as a time-out for valid cycle reception and a waiting state is entered. If a valid CS value (CSrc=next(CSopr)) is received the Valid flag is set to indicate that a valid sequence has been established and a valid sequence state entered. If a valid cycle is not received TNS expiry initiates recovery action.

TVS is used in conjunction with the Scrub and Valid flags to assist in the scrubbing of stale monitor ranks and Sequence numbers. Whenever a correct rank or sequence has been received the Scrub flag is set and TVS is reset to T_Hold. The Scrub set, Valid reset condition effectively prevents incorrect ranks and sequences being propagated through the station. Incorrect ranks are filtered by the Repeat flag being reset, incorrect sequences are scrubbed as CS_actions sets C2opr to R for out of sequence CS values. TVS expiry, due to persistent errors, results in the Scrub flag being cleared. At this point the ring has been scrubbed of stale monitor ranks and Sequence numbers. In this scrubbed state latching of ranks and Sequences is enabled to allow the ring to be resynchronized by a monitor station. TVS is reset to T_err as a time-out for valid cycle reception. If a cycle with a good rank or sequence number is not received within this time-out then TVS expiry initiates recovery action.

CC(10a) Goto Basic: Receipt of a basic mode MAC frame, or a change of class to BASIC, or an HRC_Stop signal, all cause a transition to the BASIC state. TNS expiry with the Hold flag clear and CSopr=0 indicates that there is no ranking monitor on the ring capable of initializing the ring to hybrid mode. A transition back to basic mode is made to allow packet communication to continue.

CC(10b) Reset: The HRC_Reset signal, causes the state machine to be initialized to the BASIC state, the Class variable is reset to slave.

CC(12) Recovery Required: A monitor station detecting a ring error condition which requires recovery action enters the STANDBY state in an attempt to recover the ring. This is done through the monitor contention procedure. In general the error conditions which would result in this transition are persistent sequence errors, persistent non-arrival of cycles, or monitor contention.

Persistent sequence errors and non-arrival of cycles are generally indicated by timer expiry using the procedures described earlier in this clause.

The criteria for a recovery due to persistent sequence errors are

a) TVS expiry with the Scrub flag clear and a non zero CSopr. This indicates that no usable ranks or sequence numbers have been received for at least T_err.

OR

b) The station is a monitor with the Initialize or Sustain flag set.

AND

The Hold flag is reset indicating that CSopr latching is enabled, the station has been in the slave state for at least T_Valid, and the CSopr value is still 0 indicating that no latchable CS value has been received in this time.

OR

c) The station has a non zero CSopr: indicating that the station had previously latched a value for CSopr.

AND

TNS expires with the Hold flag clear: indicating that this sequence had not been maintained and that although CSopr latching has been enabled for at least T_Valid the station has not received any latchable ranks or sequence numbers in that time.

The criteria for a recovery due to monitor contention are

a) The station is a monitor

AND

b) The station receives a cycle with CCrc = RR and CSrc < Own rank

AND

c) The station's contend flag is set or CSrc=0

AND

d) The transition would not cause data disruption due to LAB switching or the normal Latch_Rank conditions are satisfied or $CSopr < 64$ indicating that the previously latched CS value was a rank.

The aim here is to allow stations to enter Standby on monitor contention, but to prevent stations from entering the Standby state and disrupting the ring when receiving unsynchronized cycle headers caused by bit errors on the ring.

8.4.3.3 State CC2: STANDBY (hybrid mode)

A monitor is in the STANDBY state while it is active in the monitor contention procedure. All monitors may take part in the monitor contention procedure, however only those with $rank > 0$ can become cycle master. Both packet and isochronous data transfer can take place in this state, but the quality of service may be degraded if the ring is not being synchronized.

While in this state Standby_Actions are executed: cycles are generated, with CCtx= RR and CStx = rank, and transmitted using local cycle timing, the value of CSopr is set to next(CSopr) for every cycle received.

CC(20a) Goto Basic: Receipt of a Beacon frame, or any basic mode MAC frame while the Scrub flag is clear, or a change of class to BASIC, or an HRC_Stop signal, all cause a transition to the BASIC state.

CC(20b) Reset: The HRC_Reset signal causes the state machine to be initialized to the BASIC state, all variables and timers are also reset. TNS expiry with the Hold flag clear indicates that the ring could not be initialized in hybrid mode. A transition back to basic mode is made to allow packet communication to continue.

CC(21) Backoff: A transition to the SLAVE state is triggered whenever a higher priority cycle header is received. These fall into two categories: a contention cycle containing a repeatable rank or a cycle header with a repeatable sequence number.

The criteria for a repeatable rank in STANDBY are

a) a cycle with CCrc=RR and $0<CSrc<64$

AND

b) Own Class is slave or CSrc greater than Own rank

AND

c) The Hold flag is reset indicating that the station has had the chance to fill the ring with its own rank, or the Scrub flag is reset indicating that the station is not currently scrubbing and the currently latched rank is null, or the normal Latch_Rank criteria are satisfied and the received rank is not equal to the currently latched rank (this last condition removes large, no owner, ranks from the ring).

The criteria for a repeatable sequence in STANDBY are

a) a cycle with CCrc valid and $CSrc \geq 64$

AND

b) The Hold flag is reset indicating that the station has had the chance to fill the ring with its own rank, or the Scrub flag is reset indicating that the station is not currently scrubbing and the currently latched sequence is null, or CSrc is equal to the next expected sequence as indicated by next(CSopr), or the normal Latch_Sequence criteria are satisfied.

In addition to the above criteria the Backoff transition is taken by a slave or zero rank station following the scrubbing process. In this case it is not necessary to perform the entry actions for both the Standby and the Slave states simultaneously.

CC(23) Prevailed: A transition to the RESYNCH state occurs when a station wins the monitor contention procedure.

The criteria for winning the monitor contention procedure are

a) the station has Class = monitor

AND

b) receipt of a cycle with the same rank in the CS field as in the previously received cycle (CS_valid true)

AND

c) the received CS value is equal to the stations Own rank

AND

d) The Hold flag is reset indicating that the station has been transmitting its rank for at least T_Hold therefore ensuring that a stations cycles have had a chance to propagate all the way round the ring.

A station satisfying these criteria may be sure that it has the highest rank on the ring and is therefore entitled to become cycle master.

8.4.3.4 State CC3: RESYNCH (hybrid mode)

A monitor enters the RESYNCH state if it believes it has the right to become cycle master, it follows therefore that only one monitor should enter this state at any one time. In this state the monitor attempts to synchronize the rest of the ring and establish correct sequence number operation. Packet and isochronous data transfer can take place in this state however the quality of service may be degraded as the ring may not be fully synchronized.

While in this state Resync_Actions are executed: Cycles are generated, with CCtx = RS and CStx = next(CSreq), and transmitted using local cycle timing, CSreq is incremented every 125µs. On entry to this state CSopr is set equal to next(CSreq), which is the first sequence number transmitted in this state. This defines a lower bound for the sequence numbers which are expected to be received after one T_Hold time. Also on entry TNS is reset to T_Hold and the Hold flag is set, CSopr is left unchanged until TNS expires and resets the Hold flag. Once the Hold flag is reset this station's cycles have had time to propagate all round the ring, at this point CSopr may be incremented to next(CSopr) on cycle receipt. In this way CSopr and CSreq form a moving window for expected values of CSrc.

CC(30a) Goto Basic: Receipt of a Beacon frame, or any basic mode MAC frame while the Scrub flag is clear, or removal of MONITOR class status, or an HRC_Stop signal, all cause a transition to the BASIC state.

CC(30b) Reset: The HRC_Reset signal causes the state machine to be initialized to the BASIC state. TNS expiry with the Hold and M_flag flags clear indicates that this station is unable to synchronize the ring, possibly due to multiple stations being in the RESYNCH state. A transition back to basic mode is made to give other stations the chance to initialize the ring.

CC(32) Resolution: TNS expiry with the Hold flag clear indicates that this station is unable to synchronize the ring, possibly due to multiple stations being in the RESYNCH state. The M_flag being set indicates that this station reached the RESYNCH state from the MASTER state and therefore had previously been able to synchronize the ring correctly. If the Maintain flag is set an attempt is made to recover the ring in hybrid mode by taking a transition back to the STANDBY state, so as to allow monitor contention to take place between all ranked monitors. This mechanism allows cycle master resolution when two active rings are merged.

CC(34) Synchronized: On entering the RESYNCH state a monitor waits for T_Hold so as to ensure that its cycles have had time to be repeated all round the ring. It then checks for received cycles with CCrc = RS and a sequence number within the window between that contained in CSopr (initialized to the first it transmitted on transition to RESYNCH state) and the last transmitted CS. Receipt of two consecutive correct cycles (required for CS_valid to be true) indicate that all the slave stations have synchronized correctly. A transition to the MASTER state therefore occurs on receipt of the second correct cycle header.

8.4.3.5 State CC4: MASTER (hybrid mode)

The MASTER state is the normal operational state of the cycle control process in a cycle master. Only one monitor may enter this state at any one time. Both packet and isochronous data transfer can take place in this state. In the MASTER state the cycle master is free to change the programming template according to Isochronous and Packet bandwidth requirements. The programming template should only be changed in this state.

While in this state Master_Actions are executed. Cycles are generated, with CCtx = SS and CStx = Sequence Number, and transmitted using local cycle timing. On entry to this state CSopr is initialized to CSrc to establish the correct sequence for checking received sequence numbers, also the Scrub flag is set and TVS is reset to T_Hold as a time-out for the receipt of fully synchronized cycles with CC=SS. During normal operation CSopr is incremented (mod 192) for every cycle header received and TVS is reset to T_Err on receipt of correctly synchronized cycles.

CC(40a) Goto Basic: Receipt of a basic mode MAC frame, or removal of MONITOR class status, or an HRC_Stop signal, all cause a transition to the BASIC state.

CC(40b) Reset: The HRC_Reset signal causes the state machine to be initialized to the BASIC state.

CC(43) Resynchronisation Required: Expiry of TVS with the Scrub flag clear indicates that ring synchronization has been disrupted in some way. The cycle master goes back into the RESYNCH state in order to attempt to recover synchronization. As an option, useful for large values of T_Err, recovery procedures can also be initiated by detection of a cycle slip in received cycles. The criteria for this are that two correct sequential cycles are received which do not match the expected sequence defined by CSopr (i.e. CS_valid true but CSrc ≠ next(CSopr)).

8.5 Cycle generation process

The complete cycle generation process is only required in monitor stations. It maintains control over the cycle transmission process to ensure that data transmitted onto the ring is consistent with the synchronization state of the hybrid ring. To this end it generates the CC, CS and PT fields of the transmitted cycles based on its current state and the control and header information received from cycle acquisition.

The cycle generation process contains the monitor's WBC template generation state machine. The function of this state machine is to generate both the template used for sorting data at the monitor and the programming template transmitted onto the ring. The template fields are constructed by processing programming template information received from the WBC template filter in the cycle acquisition process and WBC assignments received from SMT.

The cycle generation process also derives the 8 kHz timing used for transmitting cycles. In master mode this timing may either be based on a local frequency reference or it may be synchronized to an external 8 kHz timing reference provided by the SM_HM_CYCLE_SYNC.request primitive. In slave mode this timing information is always derived from the received cycles via the RX_Start signal.

The start of a cycle is indicated to the cycle exchange process by the TX_Start signal. Cycle exchange uses this information to generate the HI_CYCLE_SYNC.indication primitive to the I-MAC at the start of cycle data. Aborting of a cycle is indicated to the cycle exchange process by the TX_Abort signal. Cycle exchange uses this information to generate HP_Invalid and HI_Invalid signals.

A non-monitor station is not required to implement the LAB, 8 kHz timing, and template generation processes. It is required to implement the cycle generation state machine, except for transitions CG(23b) and CG(23c).

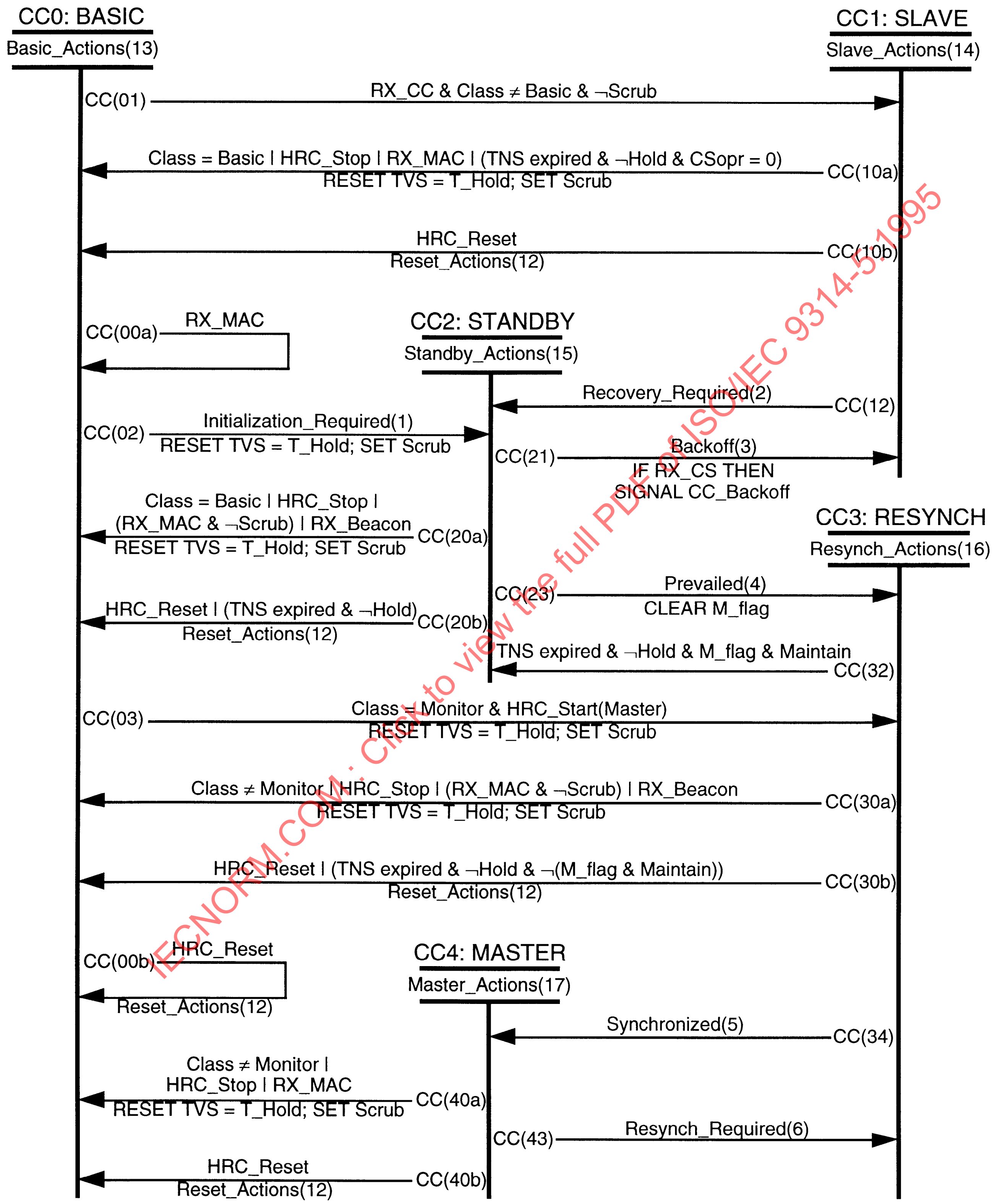


Figure 16 — HRC cycle control state diagram (Part 1 of 5)

Conditions:

(1) Initialization_Required:

Class = monitor &
((Initialize & TNS expired) | {no cycles or MAC frames on ring}
(HRC_Start(Contend)) {SMT request for hybrid mode contention}

(2) Recovery_Required:

(TVS expired & ¬Scrub & CSopr > 0) |
(¬Hold &
((CSopr = 0 & Class = monitor & (Initialize | Sustain)) |
(CSopr > 0 & TNS expired))) | {persistent cycle sequence errors}
(RX_CS & RX.CCrc = RR & RX.CSrc < Rank & Class = monitor &
(Contend | RX.CSrc = 0) &
(LAB not engaged in Standby | LAB remains engaged on Backoff |
Latch_Rank | CSopr < 64)) {monitor contention}

(3) Backoff:

(¬Scrub & (Class = slave | (Rank = 0 & ¬(Initialize | Sustain)))) |
{backoff after scrubbing if unranked}
{optional – potential immediate return to slave can be deferred}
(RX_CS & RX.CCrc = RR & RX.CSrc < 64 &
(RX.CSrc > Rank | (Class = slave & 0 < RX.CSrc < Rank)) &
(¬Hold | (¬Scrub & CSopr = 0) | (Latch_Rank & RX.CSrc ≠ CSopr))) |
{backoff to repeatable rank}
(RX_CS & RX.CCvalid & RX.CSrc ≥ 64 &
(¬Hold | (¬Scrub & CSopr = 0) |
(¬M_flag & (RX.CSrc = next(CSopr) | (RX.C2rc = S & (CSopr < 64 | ¬Scrub | Valid))))))
{backoff to repeatable sequence number}
{optional – can substitute Latch_Sequence for
(RX.C2rc = S & (CSopr < 64 | ¬Scrub | Valid))}

(4) Prevailed:

Class = monitor & RX_CS & RX.CSvalid & RX.CSrc = Rank & ¬Hold

(5) Synchronized:

Class = monitor &
RX_CS & RX.CCrc = RS & ¬RX.Cycle_Error & RX.CSvalid & ¬Hold &
((CSopr < RX.CSrc ≤ CSreq) |
(CSreq < CSopr < RX.CSrc) |
(RX.CSrc ≤ CSreq < CSopr))

(6) Resynch_Required:

Class = monitor &
((TVS expired & ¬Scrub) { | {persistent cycle sequence errors}
(Class = monitor & RX_CS & RX.CSvalid & RX.CSrc ≠ next(CSopr)))} {cycle slip}
{optional – can ignore if T_Err is small}

Figure 16 — HRC cycle control state diagram (Part 2 of 5)

(7) Repeat_Sequence:

RX_CS & RX.CCvalid &
(RX.CSrc ≥ 64 I
(RX.CCrc = RR &
(RX.CSrc > Rank I (0 < RX.CSrc < Rank & ¬Contend)) &
(RX.CSrc ≥ CSopr I ¬Scrub I Valid { I ¬RX.CSvalid I ¬Repeat })))
{ ¬RX.CSvalid I ¬Repeat is optional }

(8) Latch_Sequence:

RX_CS & RX.CSvalid & RX.CSrc ≥ 64 &
(RX.CSrc = next(CSopr) I CSopr < 64 I ¬Scrub I Valid)

(9) Latch_Rank:

RX_CS & RX.CSvalid & RX.CSrc < 64 & RX.CSrc ≠ Rank &
(RX.CSrc ≥ CSopr I ¬Scrub I Valid)

Actions:

(10) TVS_Actions:

In all states,
WHEN TVS expired & Scrub
THEN RESET TVS = T_Err; {T_Err > C_Err + 2 cycles}
CLEAR Scrub

(11) TNS_Actions:

In all states,
WHEN TNS expired & Hold
THEN IF State = slave & CSopr ≥ 64
THEN RESET TNS = T_Valid {T_Valid > C_Max + C_Err + 2 cycles}
ELSE RESET TNS = T_Recov
{T_Recov > max(any other T_Valid or (T_Hold + T_Err)) +C_Err + 2 cycles}
CLEAR Hold

(12) Reset_Actions:

RESET TVS = T_Valid;
CLEAR Scrub;
IF Class = monitor
THEN SET Class = slave

(13) Basic_Actions:

ON entry DO
SET CCopr = RR;
CLEAR CSopr;
RESET TNS = T_Valid;
CLEAR Hold, Valid, M_flag

Figure 16 — HRC cycle control state diagram (Part 3 of 5)

(14) Slave_Actions:

```
ON entry DO
    CLEAR CSopr, Repeat;
    RESET TNS = T_Valid;
    CLEAR Valid, M_flag;
    SET Hold
FOR EACH RX_CC DO {Process each received CC field}
    CC_Actions
FOR EACH RX_CS DO {Process each received CS field}
    CS_Actions;
    IF ¬Repeat_Sequence
        THEN CLEAR Repeat;
            SET CSopr = next(CSopr) {advance CSopr}
        ELSE SET Repeat;
            IF ¬(CSopr = 0 | Latch_Sequence | Latch_Rank)
                THEN SET CSopr = next(CSopr) {advance CSopr}
                ELSE {may latch this sequence but must not repeat if it returns}
                    IF ¬(Latch_Sequence | Latch_Rank)
                        THEN SET CSopr = next(CSopr) {advance CSopr}
                        ELSE {must latch this CSopr}
                            IF RX.CSrc ≠ next(CSopr)
                                THEN {start new sequence}
                                    RESET TNS = T_Hold; {T_Hold >C_Max}
                                    CLEAR Valid;
                                    SET Hold
                                ELSE {continue old sequence}
                                    IF ¬Hold
                                        THEN IF CSopr ≥ 64
                                                THEN RESET TNS = T_Valid
                                                    SET Valid
                            SET CSopr = RX.CSrc; {latch CSopr}
                    RESET TVS = T_Hold;
                    SET Scrub {prevent repeating of bad rank or sequence}
```

(15) Standby_Actions:

```
ON entry DO
    RESET TNS = T_Hold;
    SET Hold
FOR EACH RX_CC DO {Process each received CC field}
    CC_Actions
FOR EACH RX_CS DO {Process each received CS field}
    CS_Actions;
    CLEAR Repeat;
    SET CSopr = next(CSopr) {advance CSopr}
```

Figure 16 — HRC cycle control state diagram (Part 4 of 5)

(16) Resynch_Actions:

```
ON entry DO
    SET CSopr = next(CSreq);
    RESET TNS = T_Hold;
    CLEAR Valid;
    SET Hold
FOR EACH RX_CC DO {Process each received CC field}
    CC_Actions
FOR EACH RX_CS DO {Process each received CS field}
    CS_Actions;
    CLEAR Repeat;
    IF ¬Hold {advance CSopr after C_Max}
        THEN SET CSopr = next(CSopr)
```

(17) Master_Actions:

```
ON entry DO
    SET next(CSopr) = RX.CSrc;
    RESET TVS = T_Hold;
    SET Scrub;
    { CLEAR Hold, Valid } {already clear at this point}
FOR EACH RX_CC DO {Process each received CC field}
    CC_Actions
FOR EACH RX_CS DO {Process each received CS field}
    CS_Actions;
    IF RX.CSrc = next(CSopr)
        THEN SET Repeat;
            IF CCopr = SS
                THEN RESET TVS = T_Err;
                    CLEAR Scrub;
                    SET M_flag
        ELSE CLEAR Repeat
    SET CSopr = next(CSopr) {advance CSopr}
```

(18) CC_Actions:

```
FOR EACH RX_CC DO {Process each received CC field}
    IF C1rc = S & ¬Cycle_Error
        THEN SET CCopr = SR
        ELSE SET CCopr = RR;
            IF C1rc = S
                THEN INC Sync_Error_ct
```

(19) CS_Actions:

```
FOR EACH RX_CS DO {Process each received CS field}
    IF Latch_Sequence
        THEN SET C2opr = S
        ELSE IF C2rc = S & ¬RX.H_flag
                THEN INC Seq_Error_ct
```

Figure 16 — HRC cycle control state diagram (Part 5 of 5)

8.5.1 Cycle generation process state descriptions

This subclause specifies the states and state transitions of the cycle generation state machine as shown in figure 17.

8.5.1.1 State CG0: IDLE

When the cycle generation process is initialized by the HRC_Reset signal it enters the IDLE state. In this state received data is discarded and replaced by Idle symbols. This state is used to scrub cycles and fragments of cycles in basic mode, and to remove any partially concatenated frame on entry to basic mode.

CG(00) On receipt of either a RX_Abort signal or an HRC_Reset, the cycle generation state machine resets itself while in the CG0 IDLE state.

CG(02) On change of cycle control state from BASIC, the cycle generation state machine initiates the transmission of hybrid mode preamble by making a transition to the CG2 PREAMBLE state.

CG(03) Receipt of the RX_Start signal causes the cycle generation state machine to process and repeat a starting delimiter sequence by making a transition to the CG3 HEADER state.

8.5.1.2 State CG1: REPEAT

The cycle generation process enters the REPEAT state to repeat MAC PDUs in basic mode. In this state the cycle exchange process passes HP_UNITDATA.request primitives from P-MAC directly to PHY. I-MAC data is ignored and no cycles are processed.

CG(10) On receipt of either a RX_Abort signal or an HRC_Reset, the cycle generation state machine aborts the repeating of basic mode data by making a transition to the CG0 IDLE state.

CG(12) On change of cycle control state from BASIC, the cycle generation state machine initiates the transmission of hybrid mode preamble by making a transition to the CG2 PREAMBLE state.

CG(13) On receipt of the RX_Start signal from the Receive state machine the cycle generation state machine enters the CG3 HEADER state and commences transmission of a hybrid cycle beginning with the start delimiter JK.

8.5.1.3 State CG2: PREAMBLE

In this state the cycle generation state machine forces cycle exchange to send PH_UNITDATA.request(I) primitives to the local PHY entity. Under normal circumstances this results in preamble being transmitted. During normal hybrid operation the cycle generation state machine would stay in this state for between four and six symbol periods between cycles; however, during error conditions short or long preambles may be transmitted.

CG(20) The HRC reset signal causes the state machine to be initialized to the IDLE state; all associated variables and timers are also reset. The IDLE state is also entered if the cycle control state changes to Basic.

CG(23a) On receipt of the RX_Start signal from the Receive state machine the cycle generation state machine in a normally operating slave mode station enters the CG3 HEADER state and commences transmission of a hybrid cycle beginning with the start delimiter JK. The latency adjustment buffer is normally not engaged when this transition occurs.

CG(23b) In master mode, and in slave mode when generating multiple cycles (or with the LAB engaged) the cycle generation state machine waits in the CG2 PREAMBLE state until a nominal 5 Idle symbols have been transmitted and the Cycle_Request signal from the Timing Generator has been received, indicating that a new cycle should be started. It then enters the CG3 HEADER state and commences transmission of a hybrid cycle beginning with the start delimiter JK. Any phase deviation between the symbol clock and the Cycle request signal, due to clock tolerance, is compensated for by occasional transmission of either 4 or 6 symbols of preamble. In a byte wide implementation the cycle generate state machine would normally alternate between 4 and 6 idle symbols. The latency adjustment buffer must be engaged when this transition occurs in master mode, and it may be engaged when this transition occurs in slave mode.

CG(23c) In slave mode the cycle generation state machine normally waits in the PREAMBLE state until an RX_Start is detected (CG(23a)). However, in the absence of an RX_Start, if the Sustain flag is set and the last cycle received was correctly synchronized (C1opr=S), or when it is necessary to generate an initial cycle in the Standby cycle control state, then, once PA_Max Idle symbols have been transmitted, the monitor attempts to maintain synchronization of downstream stations by entering the CG2 HEADER state and generating its own cycles.

8.5.1.4 State CG3: HEADER

This is the state of the cycle generator during the transmission of the cycle header. On entry to this state the cycle generation state machine checks that an RX_Start has been received (or is queued in the LAB) indicating that a starting delimiter has been received and a receive data stream is available from cycle acquisition. The starting delimiter may be followed by a cycle header, a MAC PDU (frame or token) or stale data that needs to be scrubbed. If a new cycle is not received then the Generate flag is set to indicate that transmitted cycles must be generated locally. Any loss of cycles, due either to non receipt or preemptive termination of cycles is indicated in the new cycle header by setting CCtx to RR. In basic mode, this state prepares for a transition to hybrid mode if the received starting delimiter is followed by a cycle header.

In hybrid mode, the cycle generation state machine initiates transmission of a cycle by signalling TX_Start to cycle exchange. It then produces the CC and CS fields as a function of the CC state, the current CCopr and CSopr values, and the state of the Generate and Repeat flags. Once these fields have been sent to cycle exchange for transmission they are followed by the elements of the programming template field produced by the WBC template generation state machine.

In basic mode the State machine either repeats the received data or removes it from the ring (scrubing).

CG(30a) In basic mode, anything other than a valid frame control byte causes the state machine to return to the IDLE state, all associated variables and timers are also reset. The cycle generator also returns to the IDLE state if RX_Abort or RX_CC (indicating a cycle header to be scrubbed) is received while the cycle control state is BASIC. The transmitted JK is followed by a TT symbol pair to indicate to the downstream station that the transmission is being aborted.

CG(31)

CG(30b) After the CC field has been transmitted, the HRC reset signal causes the state machine to be re-initialized to the IDLE state, all associated variables and timers are also reset. The cycle generator also returns to the IDLE state if the cycle control state machine returns to the BASIC state. This can indicate that basic mode frames are being received. The partial cycle header is aborted by transmitting a JKTT symbol sequence to indicate to the downstream station that the transmission is being aborted. If this transition occurs before the cycle sequence field has been transmitted, H-MUX shall insert sufficient Idle symbols to ensure that at least four symbols have been transmitted following the previous JK, before transmitting the Abort sequence (JKTT). Returning to the IDLE state ensures that an arriving frame will not be concatenated with the aborted cycle.

CG(32) Receipt of the RX_Abort signal from the Receive state machine in hybrid mode indicates that the receiver has encountered a cycle abort condition in the received data stream. If a slave mode station does not engage its LAB it must abort cycle transmission and return to the PREAMBLE state. The partial cycle header is aborted by transmitting a JKTT symbol sequence to indicate to the downstream station that the transmission is being aborted. f this transition occurs before the cycle sequence field has been transmitted, H-MUX shall insert sufficient Idle symbols to ensure that at least four symbols have been transmitted following the previous JK, before transmitting the Abort sequence (JKTT).

CG(33) Receipt of the CC_Backoff signal from the cycle control state machine indicates that this station has yielded in the monitor contention procedure. Since the station is now a slave it must resynchronize to the received cycle, which caused the backoff, and restart cycle transmission. Similarly a slave station yields to a received cycle indicated by an RX_Start.

In a Standby station with the LAB engaged, the RX_Start is normally not presented directly to the cycle generate state machine but is enqueued in the LAB. If cycles are received normally then this RX_Start is normally taken out of the LAB immediately prior to transmission of a cycle so there is no need to yield, however if short cycles are received then the station may need to yield to an RX_Start taken from the LAB. Similarly when a station is in Standby and its LAB is not engaged it behaves as if it were a slave and yields to an RX_Start so that data is not lost.

On yielding the cycle generation state machine re-enters the CG3 Header state and commences transmission of the new cycle, beginning with the starting delimiter.

CG(34) Once the cycle header has been transmitted the cycle generator goes to the CG4 BODY state to transmit the body of the cycle.

8.5.1.4 State CG4: BODY

This is the normal state of the cycle generation state machine during the transmission of the data part of a cycle. If the LAB is not engaged, then data is simply passed from the received cycle to the cycle exchange process, with some filtering being performed to prevent propagation of invalid data. If the LAB is engaged, the cycle generation state machine is responsible for managing the output of the LAB to ensure that the correct data is always available to the cycle exchange process.

Valid isochronous data is available in WBCs with a current template state of ISO_OPERATE, and in the IMC when the Generate flag is clear. This data is taken either directly from the received cycle, or from the LAB when engaged, and passed to cycle exchange. If the current template state is ISO_INITIALIZE or the Generate flag is set in the IMC, Idle symbol pairs are output.

Valid packet data is available in WBCs with a current template state of TK_OPERATE, and in the DPG when the Generate flag is clear. This data is taken either directly from the received cycle, or from the LAB when engaged, and passed to cycle exchange. If the current template state is TK_INITIALIZE or the Generate flag is set in the DPG, packet data may be taken from the packet channel in the LAB (if available), or LI or LL symbol pairs are output.

CG(40) The HRC reset signal causes the state machine to be re-initialized to the IDLE state, all associated variables and timers are also reset. The cycle generator also returns to the IDLE state if the cycle control state is BASIC. This may indicate that basic mode frames are being received. The partial cycle is aborted by transmitting a JKTT symbol sequence to indicate to the downstream station that the transmission is being aborted. Returning to the IDLE state ensures that an arriving frame will not be concatenated with the aborted cycle.

CG(42a) Receipt of the RX_Abort signal from the Receive state machine in hybrid mode indicates that the receiver has encountered a cycle abort condition in the received data stream. If a slave mode station does not engage its LAB it must abort cycle transmission and return to the PREAMBLE state. The partial cycle is aborted by transmitting a JKTT symbol sequence to indicate to the downstream station that the transmission is being aborted.

CG(42b) Once the body of the cycle has been transmitted the cycle generator returns to the CG2 PREAMBLE state to transmit the next preamble.

CG(43) Receipt of the CC_Backoff signal from the cycle control state machine indicates that this station has yielded in the monitor contention procedure. Since the station is now a slave it must resynchronize to the received cycle, which caused the backoff, and restart cycle transmission. Similarly a slave station yields to a received cycle indicated by an RX_Start.

In a Standby station with the LAB engaged, the RX_Start is normally not presented directly to the cycle generate state machine but is enqueued in the LAB. If cycles are received normally then this RX_Start is normally taken out of the LAB immediately prior to transmission of a cycle so there is no need to yield, however if short cycles are received then the station may need to yield to an RX_Start taken from the LAB. Similarly when a station is in Standby and its LAB is not engaged it behaves as if it were a slave and yields to an RX_Start so that data is not lost.

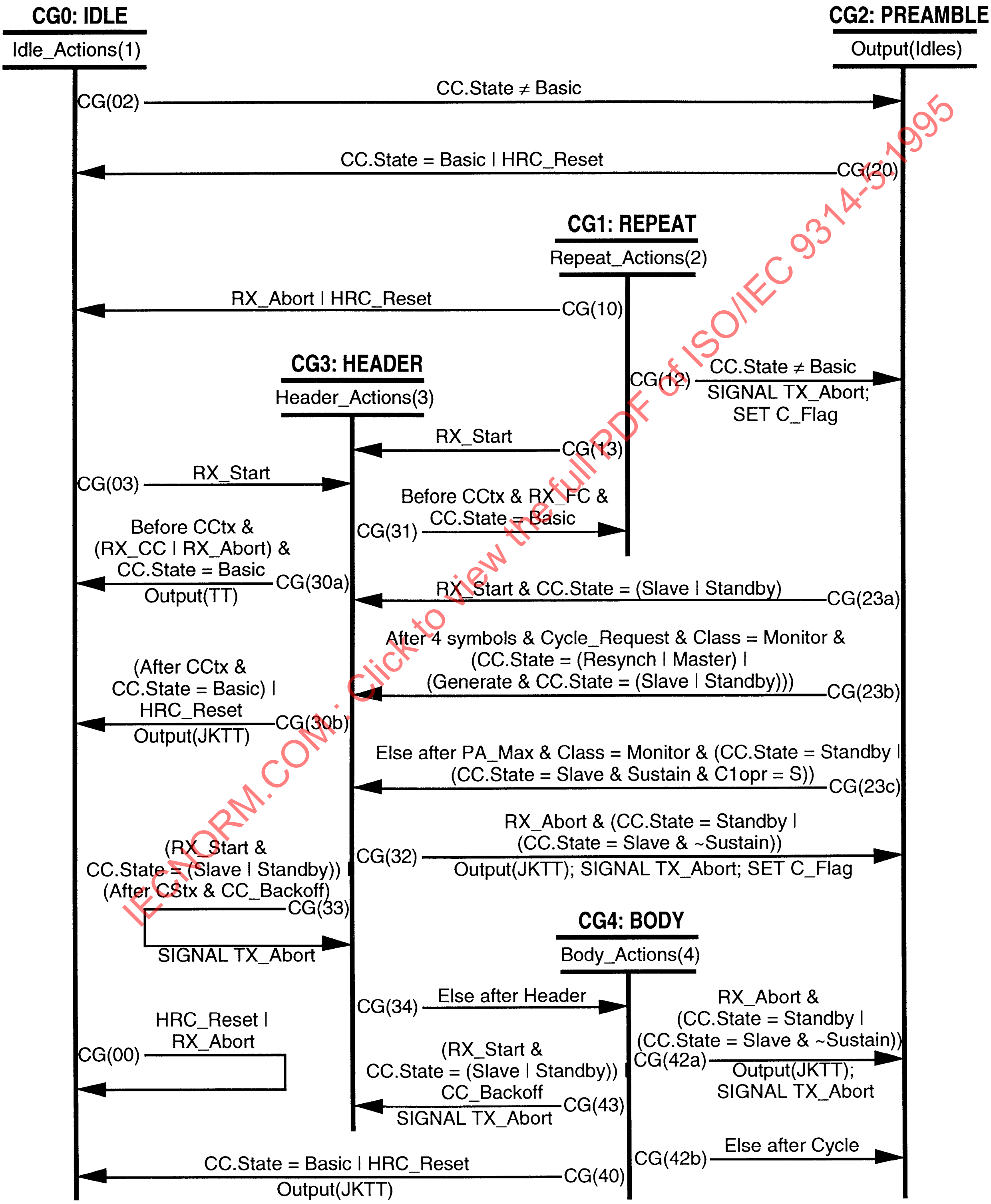


Figure 17 — HRC cycle generate state diagram (Part 1 of 4)

Actions:

(1) Idle_Actions:

ON entry DO
    SIGNAL TX_Abort; CLEAR C_Flag
REPEAT
    Output(I)

(2) Repeat_Actions:

ON entry DO
    IF C_Flag
        THEN SIGNAL TX_Abort; CLEAR C_Flag
FOR EACH Input(symbol) DO
    Output(Input(symbol))

Figure 17 — HRC cycle generate state diagram (Part 2 of 4)

On yielding the cycle generation state machine re-enters the CG3 Header state and commences transmission of the new cycle, beginning with the starting delimiter.

8.5.2 Cycle timing control

This subclause specifies the behaviour of that part of the cycle generator responsible for deriving the 8 kHz timing used by monitors for transmitting cycles in master mode (RESYNCH and MASTER states), and in Generate mode (STANDBY, and late cycles in SLAVE state).

In general there are three possible references which could be used for the generation of local 8 kHz cycle timing: a local oscillator, the SM_HM_CYCLE_SYNC.request primitive from SMT and the received cycle timing as supplied by the RX_Start signal. The reference used depends on the mode of operation and availability.

For stations operating in slave mode the local 8 kHz timing produced by timing control must be phase locked to the phase of incoming cycles. In monitors the phase of the derived timing must be such that in the event of a cycle error due to a late cycle the monitor may switch to Generate mode and commence transmission of generated cycles with the minimum possible disruption to ring synchronization in downstream stations.

In a cycle master and in Generate mode timing control locks the phase of its local 8 kHz timing to an external reference if this is supplied by the SM_HM_CYCLE_SYNC.request primitive, provision of this signal by SMT being optional. The actual (locked) phase difference between the input reference, specified by SM_HM_CYCLE_SYNC.request, and the phase of transmitted cycles is not specified.

The external reference signal can on occasion be lost, in which case this primitive may not be sent for a period of time until recovery of the external reference has been effected. In the absence of the SM_HM_CYCLE_SYNC.request primitive, or during periods of the signal being lost for some reason, timing control uses its own local oscillator to generate 8 kHz timing. Timing control shall have the capability to tolerate both the absence of the SM_HM_CYCLE_SYNC.request primitive, and the receipt of this primitive following a period of absence, without any phase jumps in the cycles transmitted on the ring.

Notwithstanding the above, certain applications may have a genuine requirement for ensuring a fixed predetermined phase between the SM_HM_CYCLE_SYNC.request primitive and transmitted cycles. In such cases provision for ensuring this fixed phase shall be provided. When such stations become cycle master phase jumps will occur in cycles transmitted on the ring following resynchronisation.

(3) Header_Actions:

```
ON entry DO
    IF (LAB not engaged & RX_Start) | {Cycle received}
     (LAB engaged & RX_Start queued) {Cycle available in LAB}
        THEN CLEAR Generate
        ELSE SET Generate; INC Generate_ct
    SIGNAL Tx_Start; Output(JK)

Before CCtx DO
    IF ~C_Flag
        THEN SIGNAL TX_Abort;
            SET C_Flag
    IF Generate | (LAB engaged & previous cycle was aborted or lost in LAB)
        THEN SET CCtx = RR     {no cycle received or cycle fragment skipped over in
                               the LAB due to it being preempted by a new cycle} .
        ELSE CASE CC.State OF
            SLAVE:
                SET CCtx = CC.CCopr
            STANDBY:
                SET CCtx = RR
            RESYNCH:
                SET CCtx = RS
            MASTER:
                IF CC.Repeat
                    THEN SET CCtx = SS
                    ELSE SET CCtx = SR

Before CStx DO
    CASE CC.State OF
        SLAVE:
            IF Generate | ¬CC.Repeat
                THEN SET CStx = 0
                ELSE SET CStx = RX.CSrc
        STANDBY:
            IF Class = Monitor
                THEN SET CStx = Rank
                ELSE SET CStx = 0
        RESYNCH, MASTER:
            SET CStx = CSreq
    IF C2tx = S
        THEN SET CSreq = next(CStx)
        ELSE SET CSreq = next(CSreq)

FOR EACH PTtx[WBC] DO
    IF Generate
        THEN SET PTtx[WBC] = Output(TG[WBC](T))
        ELSE IF TG implemented
                THEN SET PTtx[WBC] = Output(TG[WBC](PTrc))
                ELSE SET PTtx[WBC] = Output(TF[WBC](PTrc))

FOR EACH RX_Abort(current cycle) DO {current receive cycle aborted}
    SET Generate
```

Figure 17 — HRC cycle generate state diagram (Part 3 of 4)

(4) Body_Actions:

```
FOR the IMC symbol pair DO
    IF Generate
        THEN Output(II)
        ELSE IF LAB engaged
                THEN Output(Read(LAB IMC symbols))
                ELSE Output(Input(IMCsymbols))
FOR EACH DPG symbol pair DO
    IF LAB engaged
        THEN IF Generate
                THEN Output(LI or LL or Read(LAB Packet Channel symbols))
                { may optionally output filter symbols or packet data if available in LAB }
                ELSE Output(Read(LAB Packet Channel symbols))
        ELSE IF Generate
                THEN Output(LI or LL)
                ELSE Output(Input(DPG symbols))
FOR EACH WBC symbol pair DO
    CASE (IF TG implemented THEN TG.State[WBC] ELSE TF.State[WBC]) OF
        TK_INITIALIZE:
            IF LAB engaged
                THEN Output(LI or LL or Read(LAB Packet Channel symbols))
                    { may optionally output filter symbols or packet data if available in LAB }
                ELSE Output(LI or LL)
        TK_OPERATE:
            IF LAB engaged
                THEN Output(Read(LAB Packet Channel symbols))
                ELSE Output(Input(WBC symbols))
        ISO_INITIALIZE:
            Output(II)
        ISO_OPERATE:
            IF LAB engaged
                THEN Output(Read(LAB Isochronous WBC symbols))
                ELSE Output(Input(WBC symbols))
FOR EACH RX_Abort (current cycle) DO {current receive cycle aborted}
    SET Generate
```


Figure 17 — HRC cycle generate state diagram (Part 4 of 4)

8.5.3 WBC template generation

This subclause specifies the states of the WBC template generation state machine as shown in figure 18.

The major triggering event to each of the WBC template generation state machines is the receipt (from the WBC template filter state machine) of the corresponding element (symbol) of the programming template within a hybrid cycle header. The elements of this programming template identify the contents of the corresponding transmitted WBCs to be either packet or isochronous data. In a cycle master the WBC template generation state machines must also accept input from SMT, via the SM_HM_WBC.request primitive. This allows SMT to assign the programming template for use by the cycle master. The integrity of WBCs within transmitted cycles must be maintained, so although the SM_HM_WBC.request primitive may be asynchronous to cycles on the ring, any resulting events, which affect transmission of data, may only occur on a cycle boundary. Consequently SMT events may only be acted on at the start of each cycle as indicated to the state machine by the TX_Start primitive.

The states of the WBC state machines are used both in the generation of the transmitted programming template and in processing data from the LAB. In both cases the mapping between the WBC state and the appropriate action also depends on whether a station is operating in cycle master or slave mode. In general the cycle master transmits the programming template as programmed by SMT, whereas a slave normally repeats the received template. In order to maintain consistency of behaviour between slave only stations (which do not have a WBC template generation state machine) and monitor stations operating in slave mode, the states of the WBC template generation state machines in a slave must follow exactly the states of the WBC template filter state machine.

Whenever the Generate flag is set there is no valid data in the LAB. Consequently there is no valid data to be output to CXC. While this condition persists the packet channel is initialized to LI symbol pairs, and the Isochronous WBCs are initialized to II symbol pairs. The TG state machines perform an immediate transition to the appropriate Initialize states when the Generate flag is set.

8.5.3.1 State TG0: TK_INITIALIZE

When the WBC template generation process is initialized by the HRC_Reset signal it enters the TK_INITIALIZE state if Asg[WBC] is reset. During normal hybrid operation this state is entered when Asg[WBC] is reset at the beginning of a transmitted cycle, as a result of an error in the received programming template of a repeated cycle, or as a result of loss of the cycle causing Generate to become true. While in this state LI symbol pairs are forced on the LAB output for the corresponding WBC output to CXC (unless a slave station has data to repeat).

TG(00a) PT symbol error in slave: receipt of a T symbol for this WBC from the WBC template filter indicates that the symbol corresponding to this WBC in the programming template has been corrupted. Since no positive indication about the contents of this WBC has been received for this cycle the state machine stays in the TK_INITIALIZE state. A T symbol is transmitted in this field of the output programming template. The received contents of the WBC are discarded and filter bytes (LI or LL) are output to the P-MAC.

TG(00b) Incorrect PT symbol or Generating: receipt of a non R symbol for a WBC in this state by a station operating in master mode, or with its LAB engaged in slave mode, indicates that the value for this WBC transmitted by the originating station (*e.g.* as a master) has not been propagated correctly round the ring. This can be due to the symbol corresponding to this WBC in the programming template being corrupted, or due to the latency of the ring with old values being received following a PT change, or it can be due to monitor contention. When the Generate flag is set a valid cycle has not been received. In either case no valid data has been received for this WBC so the state machine stays in the TK_INITIALIZE state, to indicate that the received contents of the WBC are discarded. Since the station has its LAB engaged it can initialize the packet channel, so an R symbol is transmitted for this WBC in the programming template. Either the filtered contents of the packet channel are read from the LAB, or filter bytes (LI or LL) are output to the P-MAC

TG(01) Enable Packet Mode: receipt of an R symbol for this WBC in the programming template enables it for full operation in Packet Mode. The R symbol is repeated in this field of the output programming template.

TG(02) ISO-Assign: receipt of a Reset signal causes the Template Generate state machines to be re-initialized according to the WBC ASG[n] assignment flags programmed by SMT. Also in a master station the ASG flags are checked immediately prior to transmission of the next cycle and if the current mode of operation of a TG state machine does not correspond to the state of its associated ASG flag then the state machine is re-initialized according ASG assignment flag. When the ASG flag corresponding to the WBC is ISO.request then the WBC is re-initialized to the ISO_INITIALIZE state.

TG(03) Enable Isochronous Mode in slave: receipt of an S symbol for this WBC by a slave station enables it for operation in Isochronous Mode. The S symbol is repeated in this field of the output programming template. This transition also occurs in the Standby state, to allow arbitration when multiple different monitors have different templates during monitor contention, *i.e.* isochronous takes precedence over packet during monitor contention.

8.5.3.2 State TG1: TK_OPERATE

This is the normal operating state for a WBC template generation state machine corresponding to a WBC operating on packet mode. While in this state data is read out normally from the corresponding WBC output of the LAB during the cycle.

TG(10a) HRC_Reset: receipt of a Reset signal causes the template generation state machines to be re-initialized according to the WBC ASG[n] assignment flags programmed by SMT. When the ASG flag corresponding to the WBC is TK.request then the WBC is re-initialized to the TK_INITIALIZE state. The Generate condition also causes this transition.

TG(10b) PT symbol error in slave: receipt of a T symbol for this WBC from the WBC template filter indicates that the symbol corresponding to this WBC in the programming template has been corrupted. Since no positive indication about the contents of this WBC has been received for this cycle the state machine stays in the TK_INITIALIZE state. A T symbol is transmitted in this field of the output programming template. The received contents of the WBC are discarded and filter bytes (LI or LL) are output to the P-MAC.

TG(10c) Incorrect PT symbol: receipt of a non R symbol for a WBC in this state by a station operating in master mode, or with its LAB engaged in slave mode, indicates that the value for this WBC transmitted by the originating station (*e.g.* as a master) has not been propagated correctly round the ring. This can be due to the symbol corresponding to this WBC in the programming template being corrupted, or it can be due to monitor contention. No valid data has been received for this WBC so the state machine enters the TK_INITIALIZE state, to indicate that the received contents of the WBC are discarded. Since the station has its LAB engaged it can initialize the packet channel, so an R symbol is transmitted for this WBC in the programming template. Either the filtered contents of the packet channel are read from the LAB, or filter bytes (LI or LL) are output to the P-MAC

TG(11) Packet Mode: receipt of an R symbol for this WBC maintains full operation in Packet Mode. The R symbol is repeated in this field of the output programming template.

TG(12) ISO-Assign: receipt of a Reset signal causes the Template Generate state machines to be re-initialized according to the WBC ASG[n] assignment flags programmed by SMT. Also in a master station the ASG flags are checked immediately prior to transmission of the next cycle, if the current mode of operation of a TG state machine does not correspond to the state of its associated ASG flag then the state machine is re-initialized according ASG assignment flag. When the ASG flag corresponding to the WBC is ISO.request then the WBC is re-initialized to the ISO_INITIALIZE state.

TG(13) Enable Isochronous Mode in slave: receipt of an S symbol for this WBC by a slave station enables it for operation in Isochronous Mode. The S symbol is repeated in this field of the output programming template. This transition also occurs in the Standby state, to allow arbitration when multiple different monitors have different templates during monitor contention, *i.e.* isochronous takes precedence over packet during monitor contention.

8.5.3.3 State TG2: ISO_INITIALIZE

When the WBC template generation process is initialized by the HRC_Reset signal it enters the ISO_INITIALIZE state if Asg[WBC] is set. During normal hybrid operation this state is entered when Asg[WBC] is set at the beginning of a transmitted cycle, or as a result of an error in the received programming template in a repeated cycle. While in this state II symbol pairs are forced on the LAB output for the corresponding WBC output to CXC.

TG(20a) TK-Assign: receipt of a Reset signal causes the Template Generate state machines to be re-initialized according to the WBC ASG[n] assignment flags programmed by SMT. Also in a master station the ASG flags are checked immediately prior to transmission of the next cycle, if the current mode of operation of a TG state machine does not correspond to the state of its associated ASG flag then the state machine is re-initialized according ASG assignment flag. When the ASG flag corresponding to the WBC is TK.request then the WBC is re-initialized to the TK_INITIALIZE state.

TG(20b) PT symbol error in slave: receipt of a T symbol for this WBC from the WBC template filter indicates that the WBC template filter state machine is in the TK_INITIALIZE state. Since in a slave the WBC Generate state machine must track the WBC template filter state machine, the TK_INITIALIZE state is entered. A T symbol is transmitted in this field of the output programming template. The contents of the WBC can in fact be isochronous data that is in use by some other station. Therefore it is repeated unmodified.

TG(21) Enable Packet Mode in slave: receipt of an R symbol for this WBC by a slave station, enables it for operation in Packet Mode. The R symbol is repeated in this field of the output programming template.

TG(22) Incorrect PT symbol or Generating: receipt of a non S symbol for a WBC in this state by a station operating in master mode indicates that the value for this WBC transmitted by the station (as a master) has not been propagated correctly round the ring. This can be due to the element corresponding to this WBC in the programming template being corrupted, or (much more likely) due to the latency of the ring with old values being received following a PT change, or it can be due to monitor contention. When the Generate flag is set a valid cycle has not been received. In either case no valid data has been received for this WBC so the state machine remains in the ISO_INITIALIZE state, to indicate that filter symbols must be supplied to the I-MAC. An S symbol must be transmitted for this WBC in the programming template to initialize or recover the WBC.

TG(23) Isochronous Mode: receipt of an S symbol for this WBC enables full operation in Isochronous Mode. The S symbol is repeated in this field of the output programming template.

8.5.3.4 State TG3: ISO_OPERATE

This is the normal operating state for a WBC template generation state machine corresponding to a WBC operating on isochronous mode. While in this state data is read out normally from the corresponding WBC output of the LAB during the cycle.

TG(30a) TK-Assign: receipt of a Reset signal causes the WBC template generation state machines to be re-initialized according to the WBC ASG[n] assignment flags programmed by SMT. Also in a master station the ASG flags are checked immediately prior to transmission of the next cycle, if the current mode of operation of a TG state machine does not correspond to the state of its associated ASG flag then the state machine is re-initialized according ASG assignment flag. When the ASG flag corresponding to the WBC is TK.request then the WBC is re-initialized to the TK_INITIALIZE state.

TG(30b) PT symbol error in slave: receipt of a T symbol for this WBC from the WBC template filter indicates that the WBC template filter state machine is in the TK_INITIALIZE state. Since in a slave the WBC Generate state machine must track the WBC template filter state machine, the TK_INITIALIZE state is entered. A T symbol is transmitted in this field of the output programming template. The contents of the WBC can in fact be isochronous data that is in use by some other station. Therefore it is repeated unmodified.

TG(31) Enable Packet Mode in slave: receipt of an R symbol for this WBC by a slave station, enables it for operation in Packet Mode. The R symbol is repeated in this field of the output programming template.

TG(32a) HRC_Reset: receipt of a Reset signal causes the WBC template generation state machines to be re-initialized according to the WBC ASG[n] assignment flags programmed by SMT. When the ASG flag corresponding to the WBC is ISO.request then the WBC is re-initialized to the ISO_INITIALIZE state. The Generate condition also causes this transition.

TG(32b) Incorrect PT symbol: receipt of a non S symbol for a WBC in this state by a station operating in master mode indicates that the value for this WBC transmitted by the station (as a master) has not been propagated correctly round the ring. This can be due to the element corresponding to this WBC in the programming template being corrupted, or it can be due to monitor contention. No valid data has been received for this WBC so the state machine enters the ISO_INITIALIZE state, to indicate that filter symbols must be supplied to the I-MAC. An S symbol must be transmitted for this WBC in the programming template to initialize or recover the WBC.

TG(33) Isochronous Mode: receipt of an S symbol for this WBC maintains full operation in Isochronous Mode. The S symbol is repeated in this field of the output programming template.

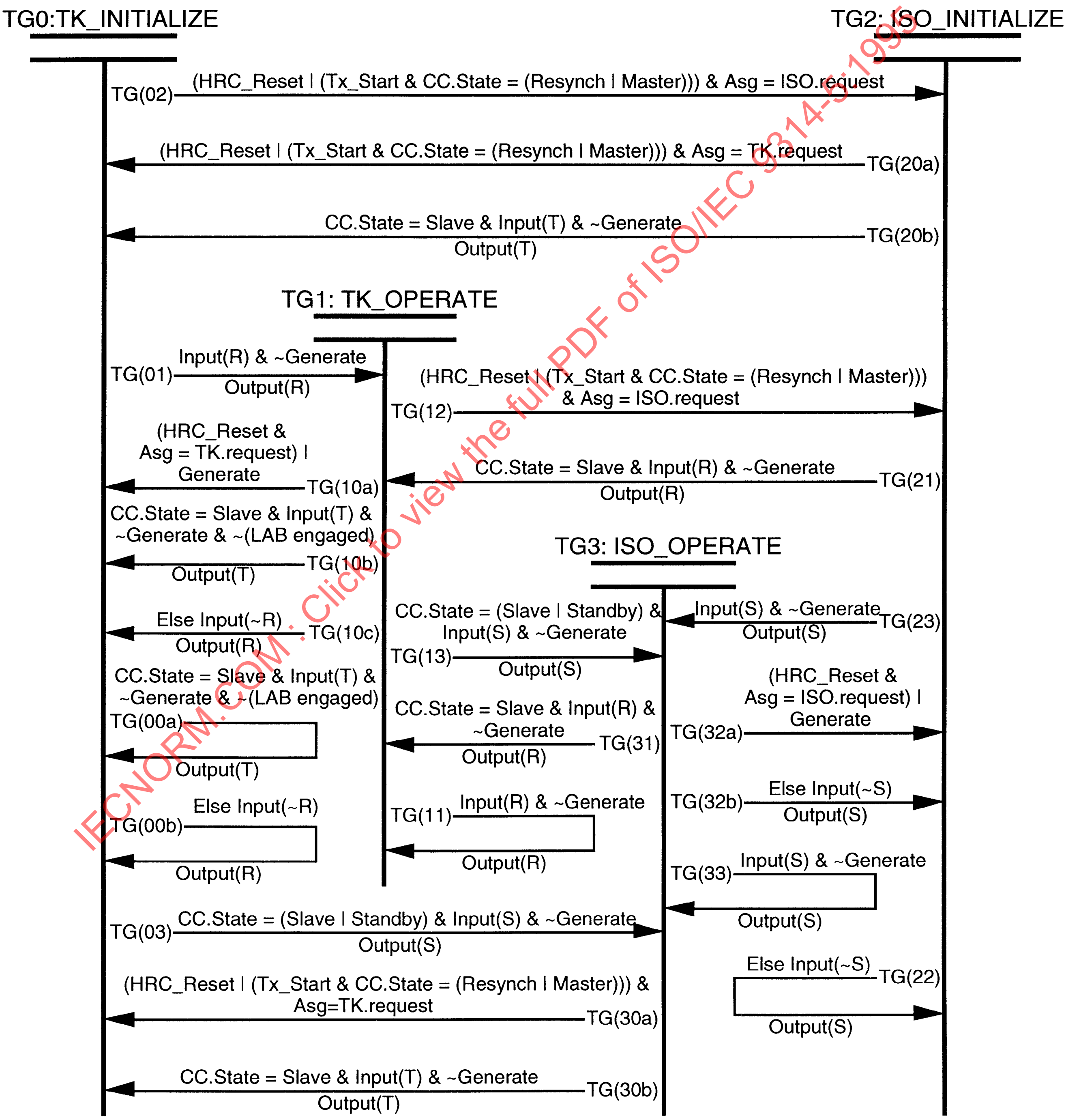


Figure 18 — WBC template generation state diagram

8.6 Cycle exchange process

The cycle exchange process contains the cycle merge and cycle distribute functions.

The cycle distribute sub-process of cycle exchange distributes the data from the input (repeated or generated) cycle to the P-MAC and the I-MAC. Data in the cyclic data groups (CDGs) is sent to the P-MAC and the I-MAC according to the programming template information in the input cycle header. In a monitor the input cycle comes from cycle generation, whereas in a non-monitor it comes directly from cycle acquisition.

The cycle merge sub-process merges the unchanged data from the input cycle header and closed WBCs with the new data from the P-MAC and the I-MAC according to the programming template and the open SAP list. The resulting data stream forms complete cycles, which are transmitted to PHY with the PH_UNITDATA.request primitive.

Depending on the implementation, it can be necessary to insert a small delay between the cycle distribute and cycle merge sub-processes, as shown in figure 13, to match the internal CXC repeat delay with the delay between data indication and data request in the P-MAC and I-MAC. This delay should be kept to a minimum as it adds to the total ring latency.

In addition to the above functions cycle exchange also generates the HI_CYCLE_SYNC.indication primitive which passes 8 kHz timing information to I-MAC. During normal synchronized operation in hybrid mode timing control sends HI_CYCLE_SYNC.indicate to I-MAC at 125 µs intervals as defined by the 8 kHz timing used for transmission of hybrid cycles. In a monitor the HI_CYCLE_SYNC.indication is synchronized to the TX_Start primitive received from cycle generation, while in a slave HI_CYCLE_SYNC.indication is synchronized to the RX_Start primitive. During hybrid synchronization the time intervals between the start of cycle transmission may vary due to resynchronisation to a received cycle: this will be reflected in the periodicity of the HI_CYCLE_SYNC.indicate primitive. Provision of this primitive in basic mode is optional.

8.6.1 Cycle distribute processing

This subclause specifies the processing required on the data in the cycle by the cycle distribute process.

Cycle distribute data actions:

```
FOR the IMC DO
    IF Open[IMC]
        THEN HI_UNITDATA.indication (input(symbol))
        ELSE HI_INVALID.indication

FOR the DPG DO
    IF Open[PSAP]
        THEN HP_UNITDATA.indication (input(symbol))

FOR EACH WBC DO
    CASE OF PTtx[WBC]
        R, T:   HI_INVALID.indication;
                IF Open[PSAP]
                    THEN HP_UNITDATA.indication (input(symbol))
        S:      IF Open[ISAP[WBC]]
                    THEN HI_UNITDATA.indication (input(symbol))
                    ELSE HI_INVALID.indication
```

```
FOR EACH TX_Abort DO
    HP_INVALID.indication;
    HI_INVALID.indication
```

8.6.2 Cycle merge processing

This subclause specifies the processing required on the data in the cycle by the cycle merge process.

Cycle merge data actions:

```
FOR EACH TX_Start DO
    PH_UNITDATA.request (input(symbol))

FOR the IMC DO
    IF Open[IMC]
        THEN PH_UNITDATA.request (HI_UNITDATA.request)
        ELSE PH_UNITDATA.request (input(symbol))

FOR the DPG DO
    IF Open[PSAP]
        THEN PH_UNITDATA.request (HP_UNITDATA.request)
        ELSE PH_UNITDATA.request (input(symbol))

FOR EACH WBC DO
    CASE OF PTtx[WBC]
        R, T:   IF Open[PSAP]
                    THEN PH_UNITDATA.request (HP_UNITDATA.request)
                    ELSE PH_UNITDATA.request (input(symbol))
        S:      IF Open[ISAP[WBC]]
                    THEN PH_UNITDATA.request (HI_UNITDATA.request)
                    ELSE PH_UNITDATA.request (input(symbol))

FOR EACH end of cycle or TX_Abort DO
    PH_UNITDATA.request (input(symbol))
```

Annex A
(informative)

Examples of the circuit-switch service class

A.1 General description

A.1.1 Functions

The circuit-switch multiplexer (CS-MUX) provides circuits as a service, and depending on the application, supports some or all of the following functions:

a) Multiplex circuits from different external sources.

b) Switch incoming circuits to associated transmission channels within a WBC.

c) De-multiplex circuits from the associated channels within the WBCs to the external sinks.

d) Perform rate adaptation between the circuit speeds and the FDDI-II data rate.

e) Provide circuit synchronization.

A.1.2 CS-MUX implementations

The circuit-switch multiplexer may be embodied within a station for the purpose of transporting internally generated isochronous signals across the FDDI-II network to another station. Basic rate ISDN voice and data communications originating at the station would be an example. Alternatively, the circuit-switch multiplexer may be bridged to a circuit-multiplexed trunk or tie-line, such as defined by CCITT G.703/G.732 (@ 2 048 kbps) or the Bell System T1 (1 544 kbps) standards. In this case, the trunk or tie line may form part of a larger public network for example, or through the use of distributed Drop and Insert Multiplexers and a local Time Slot Interchange (TSI), may be a small "isochronous-LAN" in its own right.

As shown in 5.2.3, the 6,144 Mbps wideband channel rate is a suitable common denominator for efficient multiplexing of both the 2 048 kbps CCITT and the 1 544 kbps Bell Standards. In the G.703/G.732 case, exactly 3 x 2 048 kbps lines or sources may be transparently multiplexed into a 6,144 Mbps WBC, with no stripping or modification of synchronization or signalling information being necessary. In the T1 case, exactly 4 x 1 544 kbps lines or sources may be multiplexed into a WBC by stripping the 1-bit per TDM-frame that is used for frame synchronization. Once the framing bit is stripped, the true T1 information rate of 1 536 kbps is suitable for multiplexing into a FDDI-II WBC.

The 6,144 Mbps WBC rate is well suited for multiplexing ISDN sources. Exactly 4 x 2 048 kbps Primary Rate ISDN channels or 48 x 144 kbps Basic Rate ISDN channels (2B + D) may be multiplexed into a WBC. In the latter case, the 2B channels would be multiplexed as 128 kbps isochronous channels via the I-MAC, while the D channel packets would be multiplexed into the common FDDI packet channel via the P-MAC.

A.1.3 Implementation examples

A.2 to A.4 present three techniques by which circuit-switch multiplexers may interleave information into the byte-stream defined by a transmission channel. These techniques are provided as examples only and are not intended to imply any form of standardisation. The three techniques considered are

a) Byte interleaved burst mode

b) Bit interleaved burst mode

c) Bit interleaved continuous mode

A.5 then provides an example implementation of a circuit bridge between a 2 048 kbps G.703/G.732 multiplexed line and a corresponding FDDI-II transmission channel. Again, this example is informative only, and is not intended to imply a standard implementation.

A.6 briefly describes how an FDDI-II to 1 544 kbps T1 Bridge would differ from the G.703/G.732 Bridge outlined in A.5.

A.2 CS-MUX example 1: Byte interleave burst mode

Data is transmitted and received byte-wide to and from the CS-MUX at the rate it is operated at the H-MUX, 12,5 Mbps. No Parallel to Serial and Serial to Parallel conversion and clocking adjustment (buffering) is required. Figure A.1 illustrates this architecture where the average Circuit data rate is 0,768 Mbyte/s (6,144 Mbps).

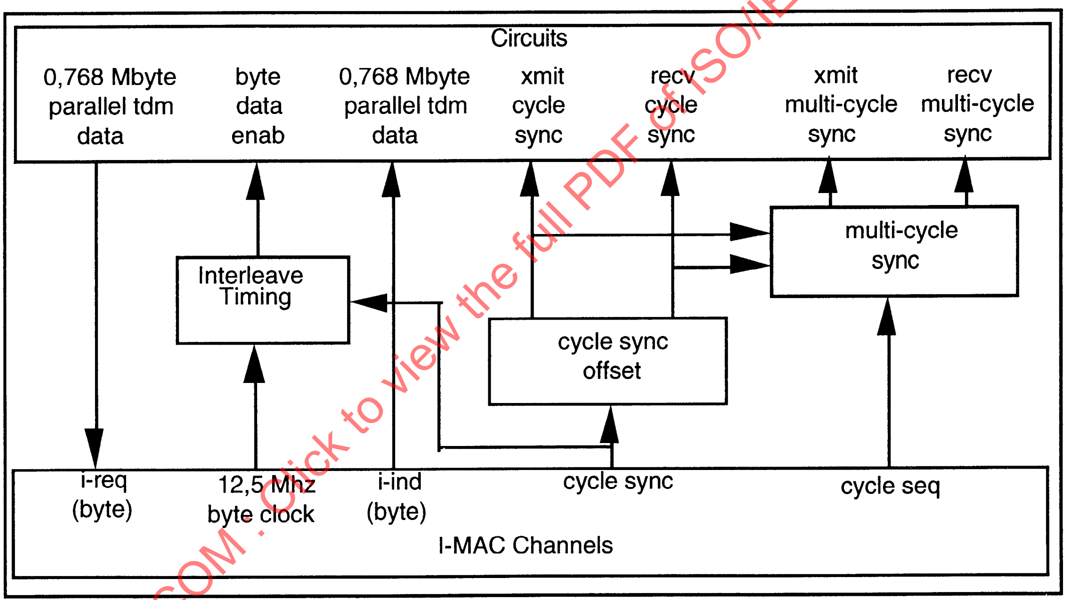


Figure A.1 — Example of a byte interleave burst mode CS-MUX

A.3 CS-MUX example 2: Bit interleave burst mode

Data is transmitted and received bit-wide to and from the CS-MUX at the instantaneous rate it is operated at the H-MUX, 12,5 Mbps. Parallel-to-serial and serial-to-parallel conversion is needed, however no buffering is used. Figure A.2 illustrates this architecture where the average Circuit data rate is 6,144 Mbps.

A.4 CS-MUX example 3: Bit interleave continuous mode

Data is transmitted and received bit-wide to and from the CS-MUX at the rate of 6,144 Mbps. Parallel to Serial and Serial to Parallel conversion and Clocking Adjustment are needed. Figure A.3 illustrates this architecture, where the average Circuit data rate is 6,144 Mbps.