IEC 61883-1

Edition 3.0 2008-02

INTERNATIONAL STANDARD

Consumer audio/video equipment – Digital interface –
Part 1: General

IEC 61883-1:2008(E)

IEC Central Office
3, rue de Varembé
CH-1211 Geneva 20
Switzerland
Email: inmail@iec.ch
Web: www.iec.ch

About the IEC

The International Electrotechnical Commission (IEC) is the leading global organization that prepares and publishes International Standards for all electrical, electronic and related technologies.

About IEC publications

The technical content of IEC publications is kept under constant review by the IEC. Please make sure that you have the latest edition, a corrigenda or an amendment might have been published.

- Catalogue of IEC publications: www.iec.ch/searchpub
The IEC on-line Catalogue enables you to search by a variety of criteria (reference number, text, technical committee,…). It also gives information on projects, withdrawn and replaced publications.

- IEC Just Published: www.iec.ch/online_news/justpub
Stay up to date on all new IEC publications. Just Published details twice a month all new publications released. Available on-line and also by email.

- Electropedia: www.electropedia.org
The world's leading online dictionary of electronic and electrical terms containing more than 20 000 terms and definitions in English and French, with equivalent terms in additional languages. Also known as the International Electrotechnical Vocabulary online.

- Customer Service Centre: www.iec.ch/webstore/custserv
If you wish to give us your feedback on this publication or need further assistance, please visit the Customer Service Centre FAQ or contact us:

Email: csc@iec.ch
Tel.: +41 22 919 02 11
Fax: +41 22 919 03 00

IEC 61883-1

Edition 3.0 2008-02

INTERNATIONAL STANDARD

Consumer audio/video equipment – Digital interface – Part 1: General

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

PRICE CODE X

ICS 33.160.01; 35.200

ISBN 2-8318-9610-X

CONTENTS

INTERNATIONAL ELECTROTECHNICAL COMMISSION

CONSUMER AUDIO/VIDEO EQUIPMENT – DIGITAL INTERFACE –

Part 1: General

FOREWORD

1) The International Electrotechnical Commission (IEC) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of IEC is to promote international co-operation on all questions concerning standardization in the electrical and electronic fields. To this end and in addition to other activities, IEC publishes International Standards, Technical Specifications, Technical Reports, Publicly Available Specifications (PAS) and Guides (hereafter referred to as "IEC Publication(s)"). Their preparation is entrusted to technical committees; any IEC National Committee interested in the subject dealt with may participate in this preparatory work. International, governmental and non-governmental organizations liaising with the IEC also participate in this preparation. IEC collaborates closely with the International Organization for Standardization (ISO) in accordance with conditions determined by agreement between the two organizations.
2) The formal decisions or agreements of IEC on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested IEC National Committees.
3) IEC Publications have the form of recommendations for international use and are accepted by IEC National Committees in that sense. While all reasonable efforts are made to ensure that the technical content of IEC Publications is accurate, IEC cannot be held responsible for the way in which they are used or for any misinterpretation by any end user.
4) In order to promote international uniformity, IEC National Committees undertake to apply IEC Publications transparently to the maximum extent possible in their national and regional publications. Any divergence between any IEC Publication and the corresponding national or regional publication shall be clearly indicated in the latter.
5) IEC provides no marking procedure to indicate its approval and cannot be rendered responsible for any equipment declared to be in conformity with an IEC Publication.
6) All users should ensure that they have the latest edition of this publication.
7) No liability shall attach to IEC or its directors, employees, servants or agents including individual experts and members of its technical committees and IEC National Committees for any personal injury, property damage or other damage of any nature whatsoever, whether direct or indirect, or for costs (including legal fees) and expenses arising out of the publication, use of, or reliance upon, this IEC Publication or any other IEC Publications.
8) Attention is drawn to the Normative references cited in this publication. Use of the referenced publications is indispensable for the correct application of this publication.
9) Attention is drawn to the possibility that some of the elements of this IEC Publication may be the subject of patent rights. IEC shall not be held responsible for identifying any or all such patent rights.

International Standard IEC 61883-1 has been prepared by technical area 4, Digital system interfaces and protocols, of IEC technical committee 100: Audio, video and multimedia systems and equipment.

This third edition of IEC 61883-1 cancels and replaces the second edition, published in 2003, of which it constitutes a technical revision.

The significant technical changes with respect to the second edition are as follows:

– allocation of a new FMT code for the 1394 Trade Association specification '601 over 1394';
– Clarification of the meaning of FMT code;
– harmonization of IEC 61883-1 with IEEE 1394.1 for speeds over S400.

The text of this standard is based on the following documents:

CDV	Report on voting
100/1236/CDV	100/1336/RVC

Full information on the voting for the approval of this standard can be found in the report on voting indicated in the above table.

This publication has been drafted in accordance with the ISO/IEC Directives, Part 2.

A list of all parts of the IEC 61883 series, under the general title *Consumer audio/video equipment – Digital interface*, can be found on the IEC website.

The committee has decided that the contents of this publication will remain unchanged until the maintenance result date indicated on the IEC web site under "http://webstore.iec.ch" in the data related to the specific publication. At this date, the publication will be

- reconfirmed,
- withdrawn,
- replaced by a revised edition, or
- amended.

A bilingual version of this publication may be issued at a later date.

CONSUMER AUDIO/VIDEO EQUIPMENT – DIGITAL INTERFACE –

Part 1: General

1 Scope and object

This part of IEC 61883 specifies a digital interface for consumer electronic audio/video equipment using IEEE 1394. It describes the general packet format, data flow management and connection management for audio-visual data, and also the general transmission rules for control commands.

The object of this standard is to define a transmission protocol for audio-visual data and control commands which provides for the interconnection of digital audio and video equipment, using IEEE 1394.

2 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

IEEE 212:2001, *Standard for a Control and Status Registers (CSR) – Architecture for microcomputer buses*

IEEE 1394:1995, *Standard for a High Performance Serial Bus*

IEEE 1394a:2000, *Standard for a High Performance Serial Bus – Amendment 1*

NOTE Throughout this document, the term "IEEE 1394" indicates a reference to the standard that is the result of the editorial combination of IEEE 1394:1995 and IEEE 1394a:2000. Devices conforming solely to IEEE 1394:1995 may conform to IEC 61883. Devices conforming to IEC 61883 should conform to IEEE 1394a:2000.

3 Abbreviations

For the purpose of this document, the following abbreviations apply.

AV/C	Audio Video Control
CHF	CIP Header Field
CIP	Common Isochronous Packet
CMP	Connection Management Procedures
CSR	Command and Status Register
CTS	Command/Transaction Set
CRC	Cyclic Redundancy Check Code
DVCR	Digital Video Cassette Recorder
EOH	End of CIP Header
FCP	Function Control Protocol
iPCR	Input Plug Control Register
iMPR	Input Master Plug Register
MPEG	Motion Picture Experts Group
oPCR	Output Plug Control Register

oMPR Output Master Plug Register

ROM Read Only Memory

spd Speed Encoding

xspd Extended Speed Encoding

For clarity, field names are shown in italics in this standard.

4 High-performance serial bus layers

4.1 Cable physical layer

All cable physical layer implementations conforming to this standard shall meet the performance criteria specified by IEEE 1394. Either the cable and connector defined in IEEE 1394:1995, or the cables and connector defined in IEEE 1394a:2000, shall be used.

When necessary for an AV device to generate a bus reset, it shall follow the requirements of IEEE 1394a:2000, 8.2.1. An AV device that initiates a bus reset should generate an arbitrated (short) bus reset, as specified by IEEE 1394a:2000, in preference to the long bus reset defined by IEEE 1394:1995.

4.2 Link layer

All link layer implementations conforming to this standard shall meet the specifications of IEEE 1394.

4.3 Transaction layer

All transaction layer implementations conforming to this standard shall meet the specifications of IEEE 1394.

5 Minimum node capabilities

A node shall conform to the following requirements.

– A node shall be cycle master capable. This is because every node has the possibility to be assigned as a root.
– A node shall be isochronous resource manager capable, as specified by IEEE 1394:1995, and shall implement the additional isochronous resource manager facilities and responsibilities specified by IEEE 1394a:2000 in 8.3.1.5, 8.3.2.3.8, 8.3.2.3.11, 8.4.2.3 and 8.4.2.6A.
– A node which transmits or receives isochronous packets shall have plug control registers (see 7.2).

5.1 Serial bus management

Bus manager capability is optional for AV devices, but, if implemented by devices conforming to this standard, shall conform to IEEE 1394.

5.2 Command and status registers

5.2.1 CSR core registers

This standard conforms to the CSR architecture. Details of its registers are specified by IEEE 1394.

The STATE_CLEAR.*cmstr* bit shall be implemented as specified by IEEE 1394a:2000, 8.3.2.2.1.

NOTE The *cmstr* bit is set automatically (see IEEE 1394a:2000, 8.3.2.2.1) by system software or hardware when a node becomes the new root after the bus reset process is completed. In this manner, it is possible to ensure the fast resumption and continuity of data transmission where the time scale is critical at the level of microseconds. The rapid activation of a new cycle master decreases the likelihood of a gap in the transmission of cycle start packets; uninterrupted transmission of cycle start packets at nominal 125 µs intervals is critical to the delivery of isochronous data within its latency requirements.

5.2.2 Serial bus node registers

Implementation requirements for bus-dependent registers in this standard conform to IEEE 1394. A node shall have the following registers:

CYCLE_TIME register

BUS_TIME register

BUS_MANAGER_ID register

BANDWIDTH_AVAILABLE register

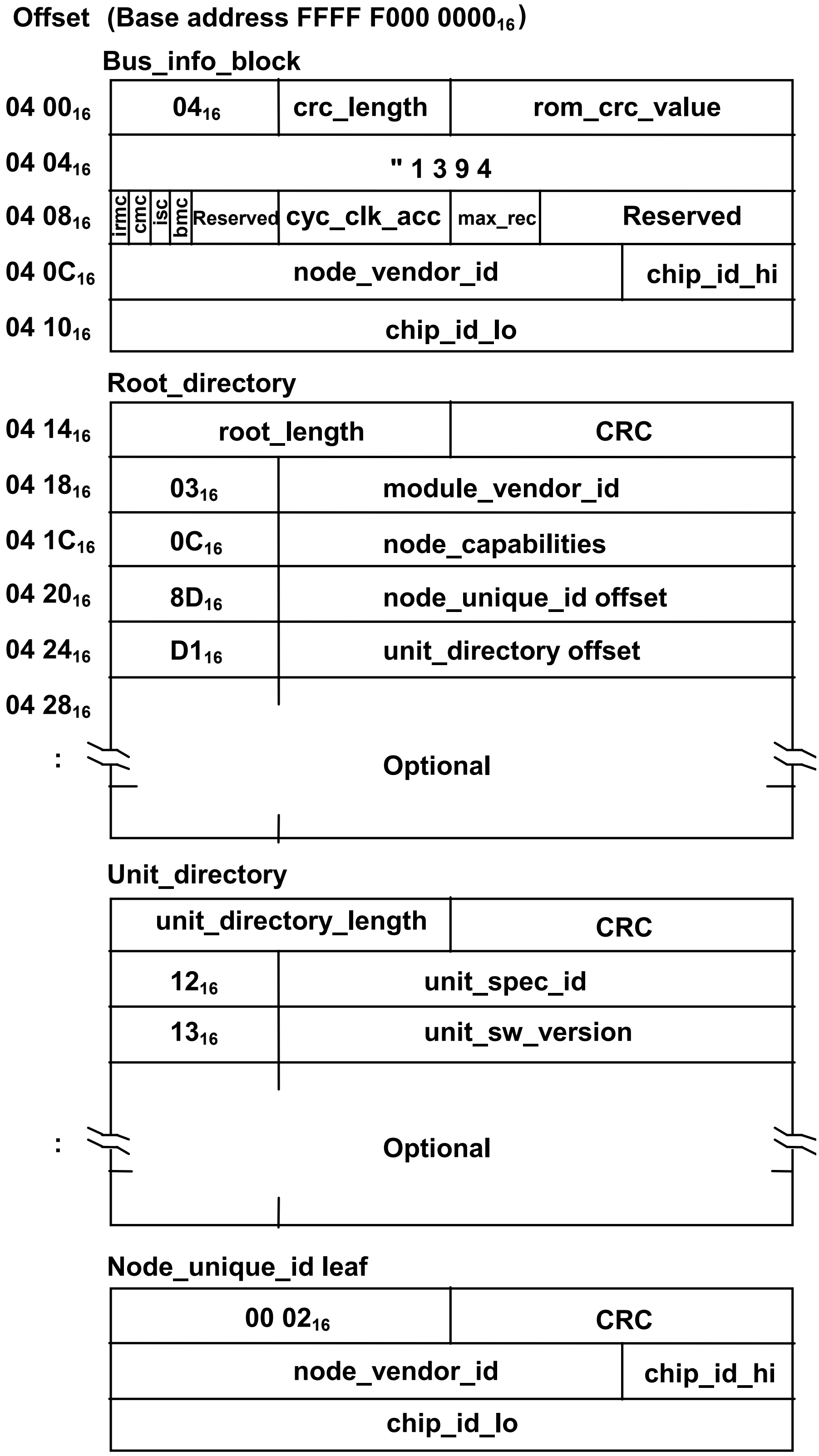
CHANNELS_AVAILABLE register

A node should have the following register specified by IEEE 1394a:2000:

BROADCAST_CHANNEL register

5.2.3 Configuration ROM requirements

A node shall implement the general ROM format as defined in IEEE 1212:2001 and IEEE 1394. Additional information required for implementations of this standard shall be included in one of the unit directories. Figure 1 shows an example of the configuration ROM implementation for this standard.



IEC 3059/02

Figure 1 – Configuration ROM

5.2.3.1 Bus_Info_Block entry

Implementation requirements for the Bus_Info_Block in this standard shall conform to IEEE 1394.

5.2.3.2 Root directory

The following entries shall be present:

– Module_Vendor_ID;

– Node_Capabilities;

– Unit_Directory (offset to a unit directory defined by this standard).

Other entries may be implemented in addition to the above required entries.

5.2.3.3 Unit directory

The following entries shall be present:

– Unit_Spec_ID;

– Unit_SW_Version.

The value of the Unit_Spec_ID and the Unit_SW_Version for this standard are given as follows:

Unit_Spec_ID:	First octet	= 00_{16}
	Second octet	= $A0_{16}$
	Third octet	= $2D_{16}$
Unit_SW_Version:	First octet	= 01_{16}

The second and third octets of Unit_SW_Version for this standard are specified in Table 1 and indicate capabilities for command/transaction sets. The Unit_SW_Version field is used to identify which protocol is supported by the device. If a device supports more than one protocol, the device shall have a separate unit directory for each protocol supported.

Table 1 – Unit_SW_Version code assignment

Unit_SW_Version	Command/transaction set
01 00 00_{16}	Reserved
01 00 01_{16}	AV/C protocol
01 00 02_{16}	Reserved for standardization by CAL
01 00 04_{16}	Reserved for standardization by EHS
01 00 08_{16}	HAVi protocol
01 00 $0A_{16}$	Automotive
01 40 00_{16}	Vendor unique
01 40 01_{16}	Vendor unique
Other values	Reserved for future standardization

6 Real time data transmission protocol

6.1 Common isochronous packet (CIP) format

6.1.1 Isochronous packet structure

The structure of the isochronous packet utilized by this standard is illustrated in Figure 2. The packet header and header CRC are the first two quadlets of an IEEE 1394 isochronous packet. The CIP header is placed at the beginning of the data field of an IEEE 1394 isochronous packet, immediately followed by zero or more data blocks.

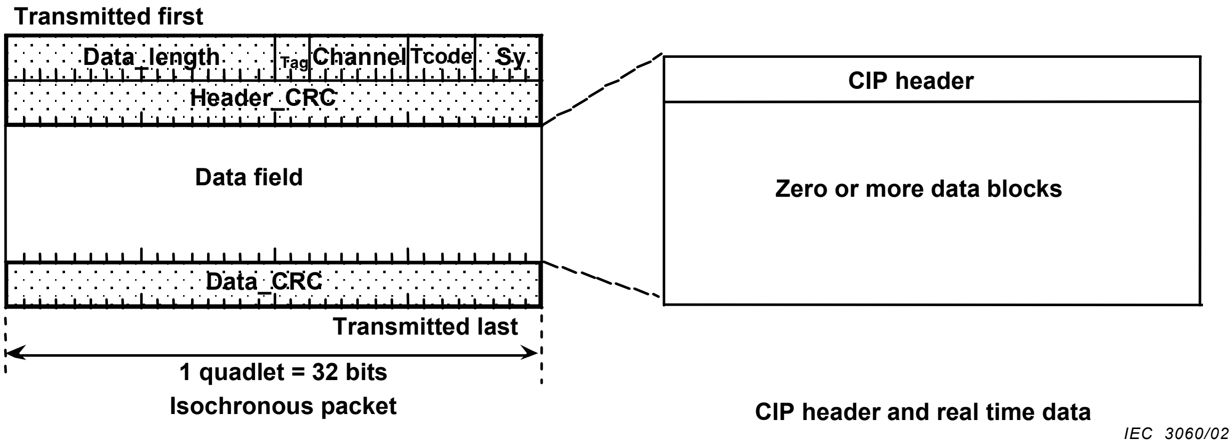


Figure 2 – Isochronous packet

6.1.2 Packet header structure

The packet header consists of the following items as specified in IEEE 1394.

Data_length: specifies the length of the data field of the isochronous packet in bytes, which is determined as follows:

CIP header size + signal data size

Tag: provides a high level label for the format of data carried by the isochronous packet

00_2 = No CIP header included

01_2 = CIP header included as specified in 6.1.3

10_2 = Reserved

11_2 = Reserved

Channel: specifies the isochronous channel number for the packet

Tcode: specifies the packet format and the type of transaction that shall be performed (fixed at 1010_2)

Sy: application-specific control field

6.1.3 CIP header structure

The CIP header is placed at the beginning of the data field of an IEEE 1394 isochronous packet. It contains information on the type of the real time data contained in the data field following it. The structure of the CIP header is shown in Figure 3.

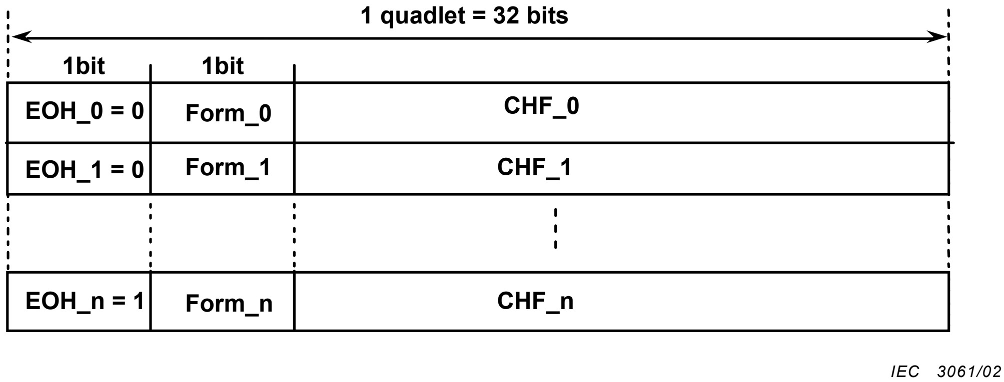


Figure 3 – CIP header

The definitions of the fields are given as follows:

EOH_n (End of CIP header): means the last quadlet of a CIP header

0 = Another quadlet will follow

1 = The last quadlet of a CIP header

Form_n: in combination with EOH, shows the additional structure of CHF_n

CHF_n (CIP header field): CIP header field of n^{th} quadlet. The additional structure of CHF_n depends on EOH_0, form_0, EOH_1, form_1, ... EOH_n, and form_n

6.2 Transmission of fixed-length source packet

This protocol transfers a stream of source packets from an application on a device to an application on other device(s). A source packet is assumed to have a fixed length, which is defined for each type of data. The data rate can be variable.

A source packet may be split into 1, 2, 4 or 8 data blocks, and zero or more data blocks are contained in an IEEE 1394 isochronous packet. A receiver of the packet shall collect the data blocks in the isochronous packet and combine them to reconstruct the source packet to send to the application.

A model conforming to these requirements is shown in Figure 4.

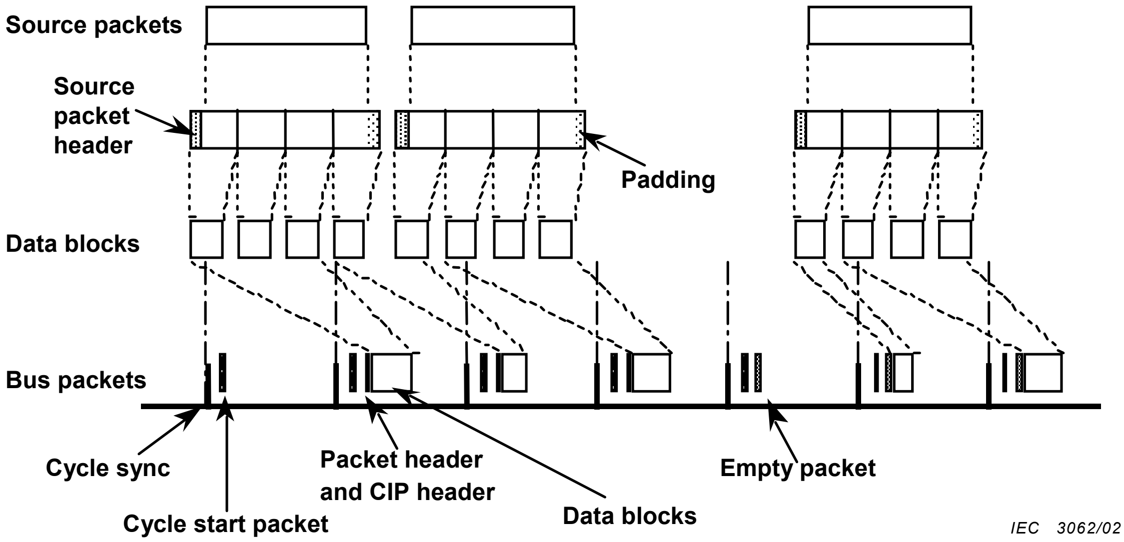


Figure 4 – Model of transmission of source packets

6.2.1 Two-quadlet CIP header (form_0=0, form_1=0)

This standard defines the two-quadlet CIP header for a fixed length source packet. There are two types for the structure of the two-quadlet CIP header as shown in Figure 5. One is the CIP header with SYT field (Figure 5a), and the other is the CIP header without SYT field (Figure 5b). If a device transmits real time data (identified by FMT) and requires time stamp in the CIP header, it shall use the SYT format.

0	0	SID	DBS	FN	QPC	SPH	rsv	DBC
1	0	0 FMT	FDF	SYT				

IEC 3063/02

Figure 5a – CIP header with SYT field

0	0	SID	DBS	FN	QPC	SPH	rsv	DBC
1	0	1 FMT	FDF					

IEC 3064/02

Figure 5b – CIP header without SYT field

Figure 5 – Two quadlets CIP header (Form_0, Form_1=0)

The definitions of the fields are given as follows.

– SID: Source node ID (node ID of transmitter)

– DBS: Data block size in quadlets

DBS field is 8 bits because 256 quadlets is the maximum payload size for S100 mode. When 8 bits are all 0, it means 256 quadlets; and 00000001_2 to 11111111_2 means 1 quadlet to 255 quadlets accordingly.

00000000_2 = 256 quadlets
00000001_2 = 1 quadlet
00000010_2 = 2 quadlets
....
11111111_2 = 255 quadlets

Several data blocks may be put into a bus packet, which is a packet to be transmitted on the bus, if a higher bandwidth is required for S200 and S400 speed.

NOTE S100, S200 and S400 are transmission speeds as defined in IEEE 1394.

– FN: Fraction number

The number of data blocks into which a source packet is divided. The allowable numbers and allocated FN codes are listed in Table 2.

Table 2 – Code allocation of FN

FN	Description
00_2	Not divided
01_2	Divided into two data blocks
10_2	Divided into four data blocks
11_2	Divided into eight data blocks

– QPC: Quadlet padding count (0 quadlet to 7 quadlets)

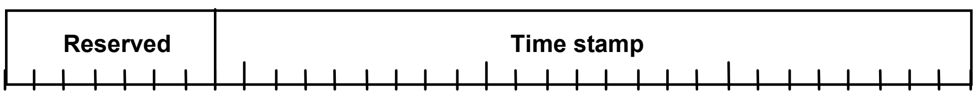
The number of dummy quadlets padded at the end of every source packet to enable division into equally sized data blocks. The value of all bits in padding quadlets is always zero.

The number of padding quadlets shall be less than the number of data blocks into which every source packet is divided, as encoded by FN.

The number of padding quadlets shall be less than the size of a single data block, as encoded by DBS. Consequently, a data block shall never consist entirely of padding quadlets.

– SPH: Source packet header

The value one indicates that the source packet has a source packet header. The format of the source packet header is shown in Figure 6. Code allocation of the time stamp field is shown in Table 3. When a time stamp is indicated, the time stamp field shall be encoded as the lower 25 bits of the IEEE 1394 CYCLE_TIME register. Other bits are reserved for future extension and shall be zeros.



IEC 3065/02

Figure 6 – Source packet header format

Table 3 – Time stamp field of source packet header

Time stamp field Higher 13 bits		Time stamp field Lower 12 bits	Description
$0\ 0000\ 0000\ 0000_2$ to $0\ 1111\ 0011\ 1111_2$	and	$0000\ 0000\ 0000_2$ to $1011\ 1111\ 1111_2$	Time stamp
$1\ 1111\ 1111\ 1111_2$	and	$1111\ 1111\ 1111_2$	No information
Other values			Reserved

– Rsv: Reserved for future extension and shall be zeros

– DBC: Continuity counter of data blocks for detecting a loss of data blocks

The value refers to the first data block following the CIP header in the bus packet. The lower FN bits contain the sequence number of the data block within its source packet. The remaining 8-FN bits form the sequence number of the source packet. The first data block of any source packet always has a sequence number with value zero. If FN is zero, then all 8 bits of DBC are used to represent a source packet sequence number. See also Table 4.

Table 4 – Placing of data block sequence

FN	Bits of DBC showing the place of data block sequence
00_2	(Not divided)
01_2	Shown in the lowest 1 bit
10_2	Shown in the lowest 2 bits
11_2	Shown in the lowest 3 bits

Table 5 – Code allocation of FMT

FMT	Description
$00\ 0000_2$	DVCR
$00\ 0001_2$	601 over 1394
$00\ 0010_2$ to $00\ 1111_2$	Reserved
$01\ 0000_2$	Audio and music
$01\ 0001_2$ to $01\ 1101_2$	Reserved
$01\ 1110_2$	Free (vendor unique)
$01\ 1111_2$	Reserved
$10\ 0000_2$	MPEG2-TS
$10\ 0001_2$	ITU-R B0.1294 System B
$10\ 0010_2$ to $10\ 1101_2$	Reserved
$11\ 1110_2$	Free (vendor unique)
$11\ 1111_2$	No data

– FMT: Format ID.

The code allocation is illustrated in Table 5.

If FMT is 111111_2 (no data), the fields for DBS, FN, QPC, SPH and DBC are ignored and no data blocks shall be transmitted. For other values of FMT, data is present and the most significant bit of the FMT field indicates whether or not a time stamp in SYT format is present. When the most significant bit of FMT is zero, the FMT-dependent field contains a time stamp in the format specified by SYT. Otherwise, the FMT-dependent field shall not contain an absolute time stamp. See also Figure 5 and Table 5.

NOTE The distinction between absolute time stamps, for example, those in the SYT format, and relative time stamps is crucial to the operation of Serial Bus bridges. Absolute time stamps require readjustment by each bridge whereas relative time stamps do not. Consult IEEE 1394.1:2004 for details.

– FDF: Format dependent field

This field is defined for each FMT.

– SYT: The code allocation of the SYT field is shown in Table 6. When a time stamp is indicated by the most significant bit of the FMT field, the SYT field shall be encoded as the lower 16 bits of the IEEE 1394 CYCLE_TIME register.

Table 6 – Time stamp of SYT field

SYT			**Description**
Higher 4 bits		**Lower 12 bits**	
0000_2 to 1111_2	and	$0000\ 0000\ 0000_2$ to $1011\ 1111\ 1111_2$	Time stamp
1111_2	and	$1111\ 1111\ 1111_2$	No information
Other values			Reserved

6.2.2 Isochronous packet transmission

Active transmitters shall send an isochronous packet in every cycle. If no data block is available, an empty packet shall be sent. An empty packet shall always contain a two-quadlet CIP header. The DBC field of an empty packet shall show the count for the first data block contained in the first non-empty IEEE 1394 isochronous packet for the same transmission stream following this empty packet. The other fields shall match the fields of the CIP header of non-empty packets on the same transmission stream.

7 Isochronous data flow management

7.1 General

To start and stop isochronous data flows on the bus and to control their attributes, the concept of plugs and plug control registers is used. Plug control registers are special purpose CSR registers.

NOTE Plugs do not physically exist on an AV device. Only the concept of a plug is used to establish an analogy with existing AV devices where each flow of information is routed via a physical plug.

This clause describes the contents of the plug control registers and how they may be modified. The set of procedures that use the plug control registers to control an isochronous data flow are called connection management procedures (CMP). The CMP that shall be used by AV devices are described in Clause 8.

7.2 Plugs and plug control registers

An isochronous data flow flows from one transmitting AV device to zero or more receiving AV devices by sending isochronous packets on one isochronous channel of the IEEE 1394 bus. An isochronous channel shall carry not more than one isochronous data flow and each isochronous data flow shall be carried on one isochronous channel.

Each isochronous data flow is transmitted to an isochronous channel through one output plug on the transmitting AV device and it is received from that isochronous channel through one input plug on each of the receiving AV devices. Each input and output plug shall not carry more than one isochronous data flow.

The transmission of an isochronous data flow through an output plug is controlled by one output plug control register (oPCR) and one output master plug register (oMPR) located on the transmitting AV device. On each AV device there is only one OUTPUT_MASTER_PLUG register for all output plugs. The OUTPUT_MASTER_PLUG register controls all attributes that are common to all isochronous data flows transmitted by the corresponding AV device. The OUTPUT_PLUG_CONTROL register controls all attributes of the corresponding isochronous data flow that are independent from attributes of other isochronous data flows transmitted by that AV device.

The reception of an isochronous data flow through an input plug is controlled by one input plug control register (iPCR) and one input master plug register (iMPR) located on the receiving AV device. On each AV device there is only one INPUT_MASTER_PLUG register for all input plugs. The INPUT_MASTER_PLUG register controls all attributes that are common to all isochronous data flows received by the corresponding AV device. The INPUT_PLUG_CONTROL register controls all attributes of the corresponding isochronous data flow that are independent from attributes of other isochronous data flows received by that AV device.

An isochronous data flow can be controlled by any device connected to the IEEE 1394 bus by modifying the corresponding plug control registers. Plug control registers can be modified by means of asynchronous transactions on the IEEE 1394 bus or by internal modifications if the plug control registers are located on the controlling device.

The use of plugs and plug control registers is illustrated in Figure 7.

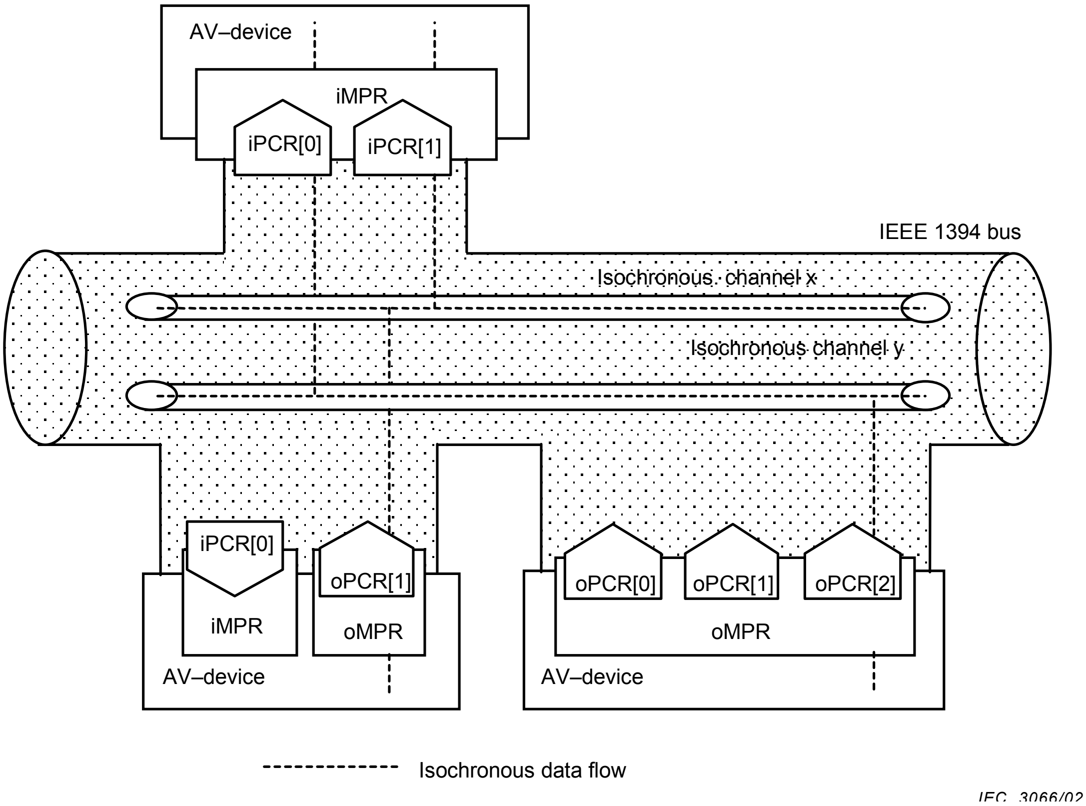


Figure 7 – Plug and PR usage

Let #iPCR and #oPCR denote the number of isochronous data flows that can be simultaneously received and transmitted respectively by an AV device (such as a multiple viewing device or a multiple tuner device). Both #iPCR and #oPCR shall be constants in the range [0 to 31] that are AV device-dependent.

Each AV device shall implement #oPCR output plugs, each controlled by one separate OUTPUT_PLUG_CONTROL register, and #iPCR input plugs, each controlled by one separate INPUT_PLUG_CONTROL register. For AV devices implementing INPUT_PLUG_CONTROL registers, a single INPUT_PLUG_CONTROL register within that AV device shall be denoted as INPUT_PLUG_CONTROL[*i*], where *i* is in the range [0 to #iPCR-1]. The INPUT_MASTER_PLUG register is optional when #iPCR = 0 and required otherwise. For AV devices implementing OUTPUT_PLUG_CONTROL registers, a single OUTPUT_PLUG_CONTROL register within that AV device shall be denoted as OUTPUT_PLUG_CONTROL[*i*], where *i* is in the range [0 to #oPCR-1]. The OUTPUT_MASTER_PLUG register is optional if #oPCR = 0 and required otherwise.

The mapping between an INPUT_PLUG_CONTROL register and an isochronous data flow in a receiving AV device, and the mapping between an OUTPUT_PLUG_CONTROL register and an isochronous data flow in a transmitting AV device, are AV device-dependent.

7.3 Connections

To transport isochronous data between two AV devices on the IEEE 1394 bus, it is necessary for an application to connect an output plug on the transmitting AV device to an input plug on the receiving AV device using one isochronous channel. The relationship between one input plug, one output plug and one isochronous channel is called a point-to-point connection. A point-to-point connection can only be broken by the same application that established it.

It is also possible that an application just starts the transmission or the reception of an isochronous data flow on its own AV device by connecting one of its output or input plugs respectively to an isochronous channel. The relationship between one output plug and one isochronous channel is called a broadcast-out connection. The relationship between one input plug and one isochronous channel is called a broadcast-in connection. Broadcast-out and broadcast-in connections are collectively called broadcast connections. A broadcast connection can be established only by the AV device on which the plug is located but it can be broken by any device. The concept of connections is illustrated in Figure 8.

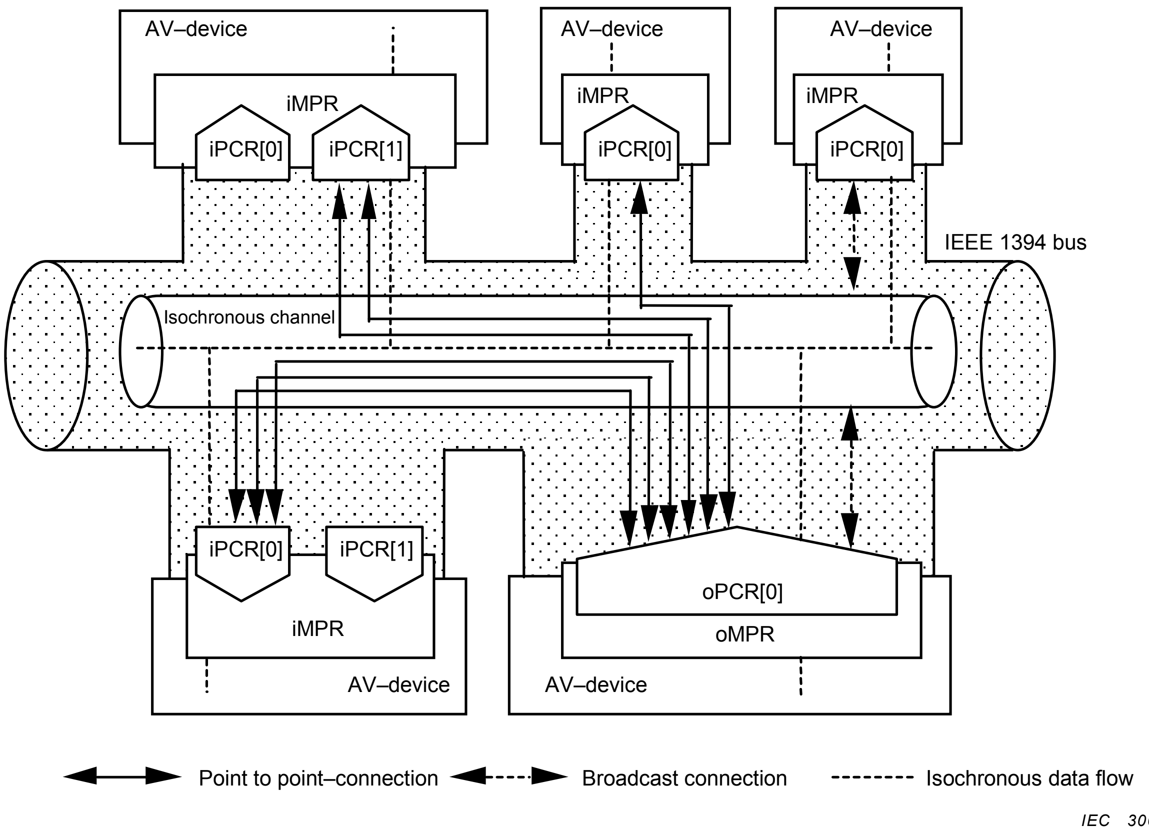
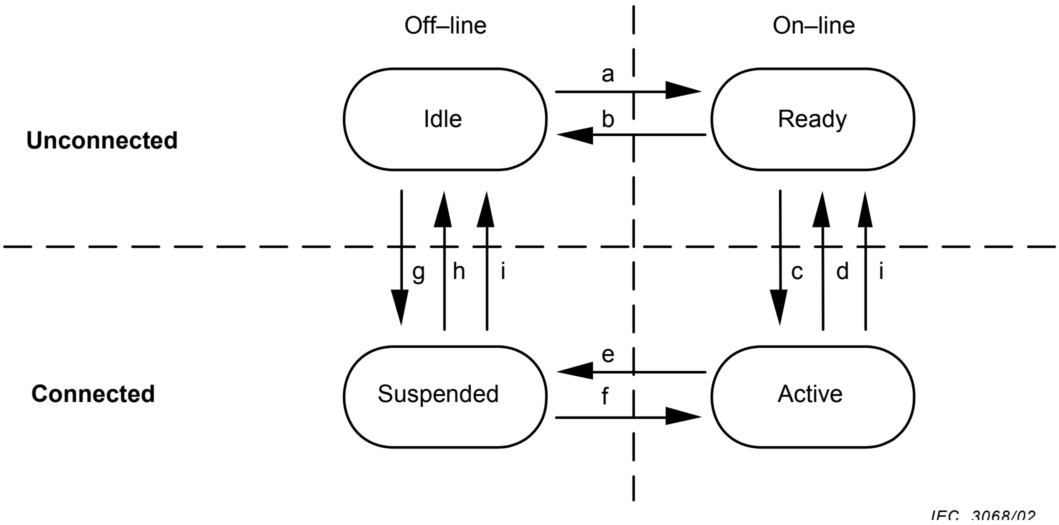


Figure 8 – Connections

Only one broadcast-out connection can exist in an output plug and only one broadcast-in connection can exist in an input plug. One broadcast connection and multiple point-to-point connections can exist simultaneously in one plug. This can be achieved by overlaying a connection over existing connections in the same input or output plug. It should be note that all connections that exist in one plug use the same isochronous channel and transport the same isochronous data flow. Multiple independent applications can create point-to-point connections between the same input and output plug.

7.4 Plug states

A plug can be in four states as described in Figure 9: idle, ready, active and suspended.



Key

a triggered internally; no action

b triggered internally; no action

c triggered by establishing the first connection; start isochronous data flow transmission/reception

d triggered by breaking the last connection; stop isochronous data flow transmission/reception

e triggered internally; suspend isochronous data flow transmission/reception

f triggered internally; resume isochronous data flow transmission/reception

g triggered by establishing the first connection; no action

h triggered by breaking the last connection; no action

i triggered by bus reset; for actions see 7.10

Figure 9 – Plug state diagram

A plug is either on-line or off-line. Only a plug that is on-line is capable of transmitting or receiving an isochronous data flow.

NOTE 1 Being capable does not mean that the plug is actually transmitting or receiving an isochronous data flow.

A plug may be off-line, for example, because it relies on resources that are (temporarily) unpowered or otherwise unavailable.

NOTE 2 The reasons that cause a plug to switch between on- and off-line are internal to the AV device on which the plug is located and do not fall within the scope of this standard.

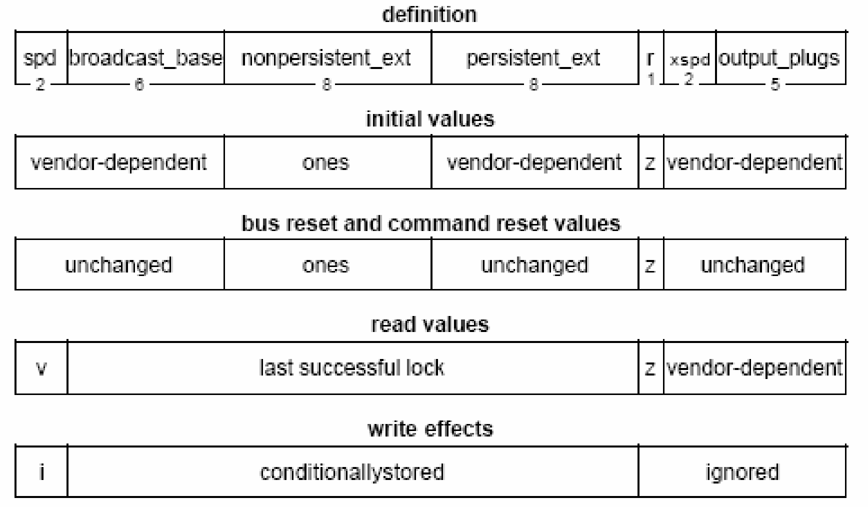
A plug to which no connections exist is called unconnected. A plug to which one or more connections exist is called connected. A plug that is connected and on-line is called active. Only an active plug shall transmit or receive an isochronous data flow except in the case of a bus reset where the isochronous data flow is resumed immediately after the bus-reset according to the procedures described in 7.10. A plug shall cease transmitting an isochronous data flow within 250 µs after becoming unconnected via transition d shown in Figure 9.

In Figure 9, all possible transitions from one state to another are given. Transitions are atomic and are effected by modifying the corresponding plug control register as described in 7.9.

NOTE 3 In order to ensure that the contents of plug registers are reliable, any intermediate results which may occur during a state transition should not be made available. A technique to achieve this is to disable access to the plug registers (for example, by masking relevant interrupt mechanisms) once a state transition is invoked, and to ensure that the state transition is completed as an indivisible process without being interrupted, suspended or modified in any way. Under these conditions, a transition is said to be atomic.

7.5 OUTPUT_MASTER_PLUG register definition

The OUTPUT_MASTER_PLUG register, as shown in Figure 10, provides information about and permits control of common aspects of a node's oPCR registers.



IEC 131/08

Figure 10 – oMPR format

The persistent extension *persistent_ext* and non-persistent extension *nonpersistent_ext* fields are defined for future extensions.

The *spd* field shall specify the maximum speed for isochronous data transmission that any of the oPCR registers may use, as specified in Table 6. When the *spd* field has a value of three, the *xspd* field shall specify the maximum speed for isochronous data transmission that any of the oPCR registers may use, as specified in Table 6. Otherwise, if *spd* is less than three, *xspd* shall be zero.

Table 1 – oMPR/iMPR/oPCR speed encoding *spd* and extended speed encoding *xspd*

spd	**IEEE 1394 speed**	***xspd***	**IEEE 1394 speed**
00_2	S100	00_2	S800
01_2	S200	01_2	S1600
10_2	S400	10_2	S3200
11_2	Maximum data rate specified by *xspd*	11_2	Reserved for future standardization

The broadcast channel base *broadcast_base* field shall specify the base channel number used to determine the channel number used for broadcast out connections. When a broadcast out connection is established for a plug for which a point-to-point connection does not simultaneously exist, the channel field of the oPCR register shall be set to 63 if *broadcast_base* equals 63 and otherwise shall be set to (*broadcast_base* + *n*) modulo 63, where *n* is the ordinal of oPCR[*n*].

The number of output plug *output_plugs* fields contains the number of output plugs an AV device implements as defined in 7.2. The *output_plugs* field shall specify the total count of oPCR registers implemented by a node. Between zero and 31 oPCR registers may be implemented. If one or more oPCR registers are implemented, they shall lie within the contiguous address range FFFF F000 0904_{16} to FFFF F000 0900_{16} + 4 x *output_plugs*, inclusive.

7.6 INPUT_MASTER_PLUG register definition

The INPUT_MASTER_PLUG register, as shown in Figure 11, provides information about and permits control of common aspects of a node's INPUT_PLUG_CONTROL registers.

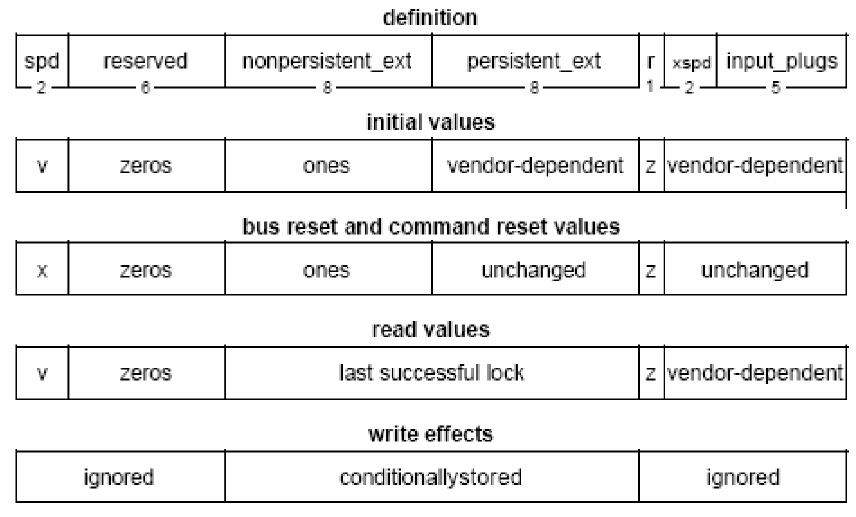


Figure 11 – iMPR format

The *spd* field shall specify the maximum speed at which any of the node's input plugs may receive isochronous data, as encoded by Table 7.

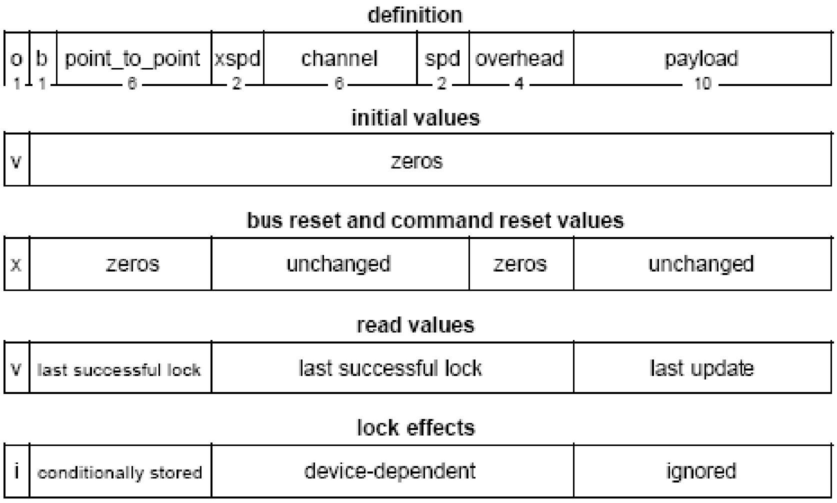
The *nonpersistent_ext* and *persistent_ext* fields are reserved for future standardization.

When the *spd* field has a value of three, the *xspd* field shall specify the maximum speed at which any of the node's input plugs may receive isochronous data, as encoded by Table 7. If *spd* is less than three, the value of *xspd* shall be zero.

The *input_plugs* field shall specify the total count of INPUT_PLUG_CONTROL registers implemented by a node, as defined in 7.2. Between zero and 31 INPUT_PLUG_CONTROL registers may be implemented. If one or more INPUT_PLUG_CONTROL registers are implemented, they shall lie within the contiguous address range FFFF F000 0984_{16} to FFFF F000 0980_{16} + 4 x *input_plugs*, inclusive.

7.7 OUTPUT_PLUG_CONTROL register definition

The format of the OUTPUT_PLUG_CONTROL register is shown in Figure 12. Each OUTPUT_PLUG_CONTROL register permits the description and control of both broadcast and point-to-point connections that originate with the associated plug. OUTPUT_PLUG_CONTROL registers shall be implemented within a contiguous address space and are referenced by an ordinal *n*, where *n* starts at zero; oPCR[*n*] refers to the register addressable at FFFF F000 0904_{16} + 4 x *n*.



IEC 133/08

Figure 12 – oPCR format

The online bit (abbreviated as *o* in the Figure 12) shall specify the on-line status of the plug resources controlled by the OUTPUT_PLUG_CONTROL register. An online bit value of zero shall indicate that the plug is off-line and not capable of transmitting isochronous data. An online value of one shall indicate that the plug may be configured and used for isochronous data transmission.

NOTE Plug status can change dynamically from off-line to on-line as device resources become unavailable or available. The causes of a change in plug status reported by the *online* bit are vendor-dependent.

The broadcast bit (abbreviated as b in Figure 12) shall specify whether a broadcast connection exists for the output plug. A value of zero indicates that no such connection exists.

The *point_to_point* field shall specify the number of point-to-point connections that exist for the output plug.

When the *spd* field has a value of three, the *xspd* field shall specify the speed to be used for isochronous data transmissions for the plug, as encoded by Table 6. Otherwise, the value of *xspd* shall be zero. If the *spd* field has a value of three and x*spd* is set to a value greater than the value of OUTPUT_MASTER_PLUG.*xspd*, isochronous data transmissions shall be disabled for the plug.

The channel field shall specify the channel number used in isochronous data transmissions for the plug.

The *spd* field shall specify the speed to be used for isochronous data transmissions for the plug, as encoded by Table 6. If *spd* is set to a value greater than the value of the *spd* field in the OUTPUT_MASTER_PLUG register, isochronous data transmissions shall be disabled for the plug.

The overhead field shall encode a value used in the calculation of the isochronous bandwidth allocation necessary for isochronous data transmissions associated with the plug, as shown in Table 8. Isochronous bandwidth is expressed in terms of bandwidth allocation units, as defined by IEEE 1394. One bandwidth allocation unit represents the time required to transmit one quadlet of data at the S1600 data rate, roughly 20 ns. If overhead is non-zero, the total bandwidth allocation necessary is expressed as overhead × 32 + (payload + 3) × 24 − (*xspd* + *spd*). Otherwise, the total bandwidth allocation can be obtained from 512 + (payload + 3) × 24 − (*xspd* + *spd*). In the preceding formulae, overhead, payload, *spd* and *xspd* represent the values of these fields in the OUTPUT_PLUG_CONTROL register.

NOTE In the formulae above, there is a negative exponent at the S3200 data rate. When dividing by two at this data rate, the result should be rounded up to the next larger integer value.

Table 8 – oPCR overhead ID encoding

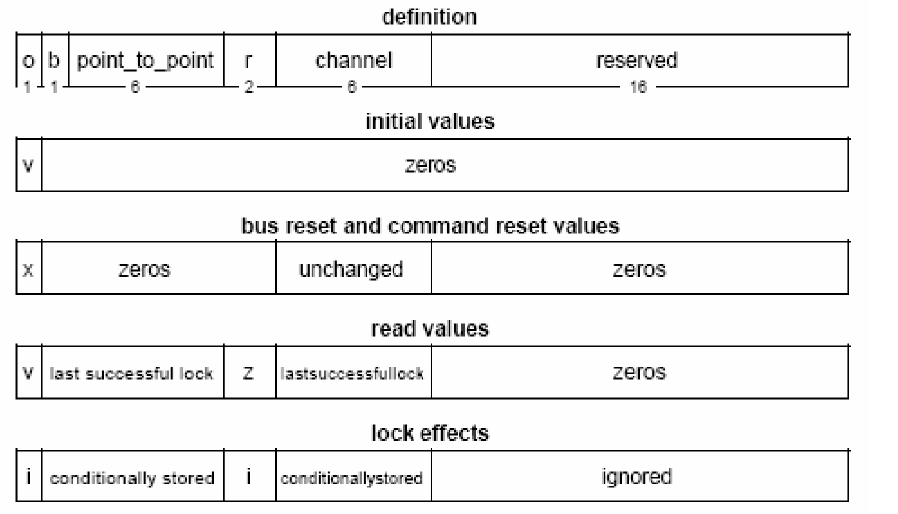
Overhead ID	IEEE 1394 bandwidth allocation units
0000_2	512
0001_2	32
0010_2	64
0011_2	96
0100_2	128
0101_2	160
0110_2	192
0111_2	224
1000_2	256
1001_2	288
1010_2	320
1011_2	352
1100_2	384
1101_2	416
1110_2	448
1111_2	480

The payload field shall specify the maximum number of quadlets that may be transmitted in a single isochronous packet for this plug. The interpretation of payload depends upon the value of OUTPUT_PLUG_CONTROL.*spd*. If *spd* is less than three, a payload value of zero indicates a maximum of 1024 quadlets; all other values represent a maximum of payload quadlets. Otherwise, if *spd* is equal to three, a payload value of zero indicates a maximum of 1024 × 2 *xspd* + 1 quadlets; all other values represent a maximum of payload × 2 *xspd* + 1 quadlets.

NOTE The value of payload does not include the isochronous header, header CRC or data CRC required as part of an isochronous packet; it counts only those quadlets that are part of the isochronous data payload.

7.8 INPUT_PLUG_CONTROL register definition

The INPUT_PLUG_CONTROL register, as given in Figure 13, permits the description and control of point-to-point connections that terminate at the associated plug. INPUT_PLUG_CONTROL registers shall be implemented within a contiguous address space and are referenced by an ordinal *n*, where *n* starts at zero; iPCR[*n*] refers to the register addressable at FFFF F000 $0984_{16} + 4 \times n$.



IEC 134/08

Figure 13 – iPCR format

The online bit (abbreviated as *o* in the Figure 13 bit shall specify the on-line status of the plug resources controlled by the INPUT_PLUG_CONTROL register. An online bit value of zero shall indicate that the plug is off-line and not capable of receiving isochronous data. An online value of one shall indicate that the plug may be configured and used for isochronous data reception.

NOTE Plug status can change dynamically from off-line to on-line as device resources become unavailable or available. The causes of a change in plug status reported by the *online* bit are vendor-dependent.

The broadcast bit (abbreviated as b in Figure 13 shall specify whether a broadcast connection exists for the input plug. A value of zero indicates that no such connection exists.

The *point_to_point* field shall specify the number of point-to-point connections that exist for the input plug.

The channel field shall specify the channel number used in isochronous data reception for the plug.

7.9 Plug control register modification rules

The contents of a plug control register shall be modified either internally by the AV device on which the plug control register is located or externally via the IEEE 1394 bus by using a quadlet compare_swap lock transaction as defined in IEEE 1394. The effect of an external modification is specified as the "lock effect" in Figures 10 to 13 and described in 7.5 to 7.8. Internal modifications shall behave as a compare_swap lock transaction as defined in IEEE 1394.

Each plug control register defined in 7.5 to 7.8 shall store any value according to the definition of write/lock effect if and only if the compare/swap lock transaction returns "resp_complete". A plug shall behave according to the requirements of 7.5 to 7.8 for the values that are stored in the plug control registers.

The following rule for modifying the contents of an INPUT_MASTER_PLUG register and OUTPUT_MASTER_PLUG register is specified:

– all modifications shall adhere to the definitions of the OUTPUT_MASTER_PLUG register and INPUT_MASTER_PLUG register as specified in 7.5 and 7.6 respectively.

The following rules for modifying the contents of an INPUT_PLUG_CONTROL register and OUTPUT_PLUG_CONTROL register are specified as follows:

– all modifications shall adhere to the definitions of the OUTPUT_PLUG_CONTROL register and INPUT_PLUG_CONTROL register as specified in 7.7 and 7.8 respectively;

– the channel and associated bandwidth (see 7.7) as stored in an OUTPUT_PLUG_CONTROL register shall be allocated during the entire time the corresponding output plug is connected;

– the channel number field and data rate field of an OUTPUT_PLUG_CONTROL register shall not be modified while the corresponding output plug is connected;

– the channel number field in an INPUT_PLUG_CONTROL register shall not be modified while its point-to-point connection counter field is not equal to zero;

– the broadcast connection counter field shall be set internally;

– when an output plug becomes connected, the data rate field, overhead_ID field, channel number field, broadcast connection counter field and point-to-point connection counter field shall be modified in the same compare_swap lock transaction;

– if the broadcast connection counter of an OUTPUT_PLUG_CONTROL register is modified from zero to one while its point-to-point connection counter remains set to zero, the channel number shall be modified in the same compare_swap lock transaction according to the formula given in 7.5.

7.10 Bus reset

When a bus reset occurs, the following actions shall be performed.

a) All AV devices that had connected input and output plugs prior to the bus reset shall continue respectively to receive and transmit the isochronous data flow immediately after the bus reset according to the values in the plug control registers immediately before the bus reset.

b) AV devices that had connected input and output plugs prior to the bus reset shall behave according to the values in the corresponding plug control registers after isoch_ resource_delay (equal to 1,0 s) following the bus reset.

7.11 Plug control register access rules

The plug control registers occupy part of a node's address space, as shown by Figure 14.

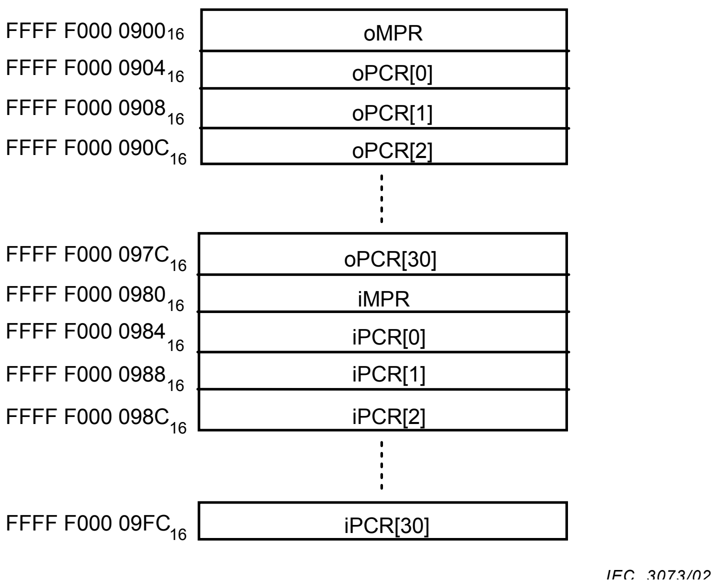


Figure 14 – PCR address map

A node that implements plug control registers shall support quadlet read requests for all implemented registers. The node shall also support lock requests for all implemented registers as long as the *destination_offset* is quadlet aligned, the *extended_tcode* is equal to two (compare and swap) and the *data_length* is equal to four. Neither quadlet write nor block write requests shall be supported for any plug control registers, whether implemented or not. If an otherwise valid request is received for an unimplemented plug control register, the node should reject the request with a response of *resp_address_error* but may complete the transaction with *resp_complete* and response data of zeros. A node may support block read requests addressed to the plug control register address space. If the combination of *destination_offset* and *data_length* for a block read request includes unimplemented plug control registers, the node may reject the request with a response of *resp_address_error*. However, if the node successfully completes the transaction, the response data returned for the unimplemented registers shall be zero.

8 Connection management procedures (CMP)

8.1 Introduction

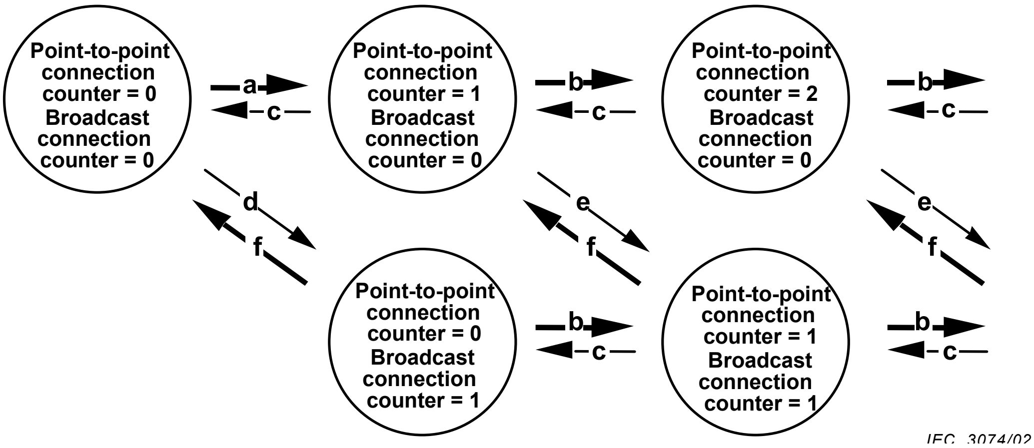
This clause describes the procedures that an application shall use to manage connections between input and output plugs of AV devices by modifying plug control registers according to the rules defined in Clause 7. Only connections as defined in Clause 7 of this standard can be managed. The following management procedures are defined for each connection type:

– establishing a connection;

– overlaying a connection;

– breaking a connection.

These operations involve the incrementing and decrementing of connection counters in the plug control registers. Figure 15 shows the relationship between these operations for the different connection types. The procedures for each connection type are described by flow diagrams in Figures 16 to 28. No change to the contents of a plug control register is executed until the first modify operation following it in the flow diagram. The flow diagrams represent possible implementations of the procedures. Other conforming implementations are possible. An implementation conforms if, and only if, it does not violate the plug control register modification rules (see 7.9) and the state transition diagram of Figure 15.



Key

a establishing a point-to-point connection; permitted by any application

b overlaying a point-to-point connection; permitted by any application

c breaking a point-to-point connection; permitted by an application that has previously established or overlaid a point-to-point connection

d establishing a broadcast connection; permitted by an application located on the device where the PCR is located

e overlaying a broadcast connection; permitted by an application located on

f breaking a broadcast connection; permitted by any application

Figure 15 – Point-to-point and broadcast connection counter modifications

8.2 Managing point-to-point connections

Point-to-point connections are protected in the sense that a point-to-point connection can only be broken by the same application that established it. Consequently, the active output plug does not stop the transmission of the isochronous data flow as long as the application does not break its point-to-point connection to that output plug.

8.2.1 Procedure for establishing a point-to-point connection

This procedure creates a protected connection between one unconnected input plug and one unconnected output plug using one unused channel. Figure 16 shows an implementation conforming to this procedure.

NOTE The choice of which OUTPUT_PLUG_CONTROL register and INPUT_PLUG_CONTROL register on the transmitting and receiving AV device respectively are used does not fall within the scope of this standard. The choice of which channel, data rate and overhead_ID are used also does not fall within the scope of this standard.

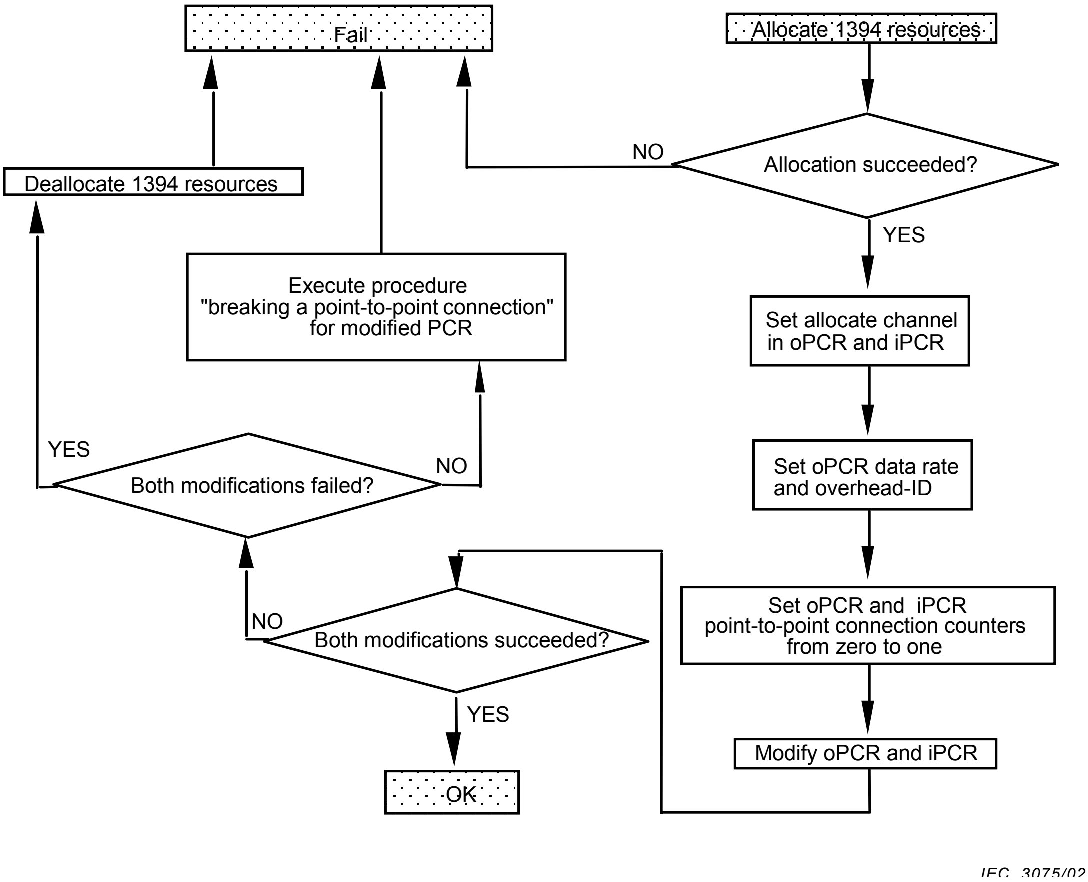


Figure 16 – Establishing a point-to-point connection

8.2.2 Procedure for overlaying a point-to-point connection

This procedure adds a protected connection to a connected output plug between that output plug and an input plug. The isochronous channel that the output plug is using to transmit the isochronous data flow shall be used for the added point-to-point connection. Figure 17 shows an implementation conforming to this procedure.

NOTE The choice of which INPUT_PLUG_CONTROL register on the receiving device is used does not fall within the scope of this standard.

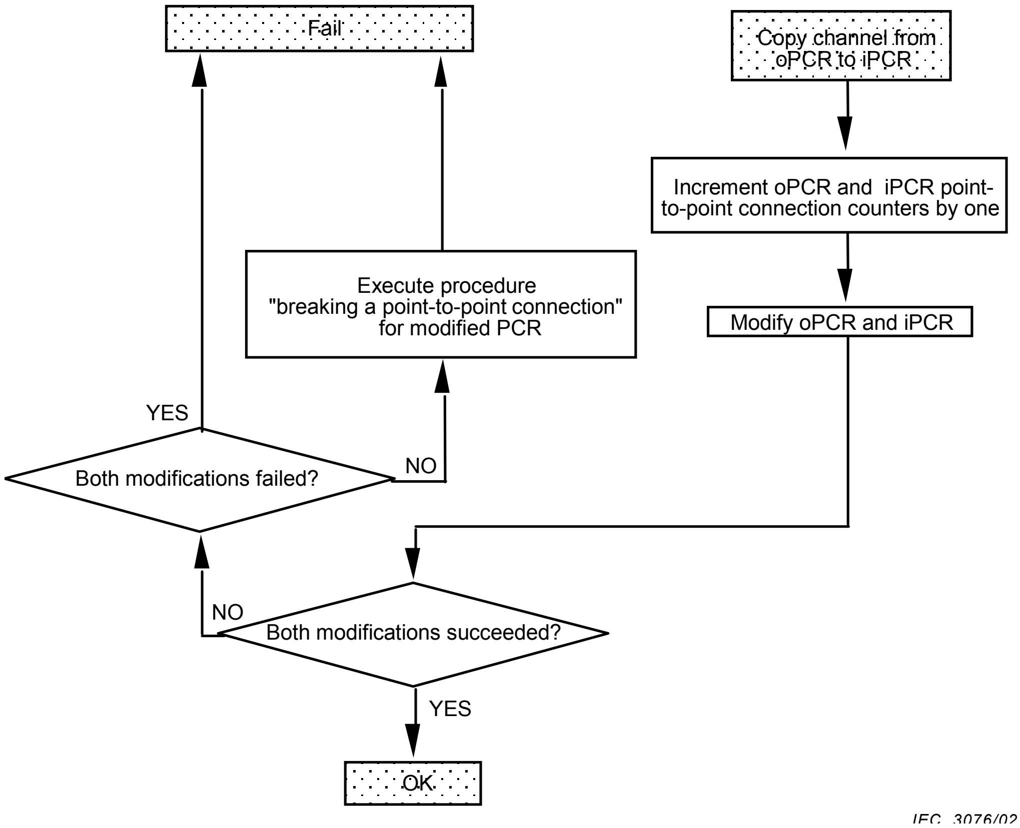


Figure 17 – Overlaying a point-to-point connection

8.2.3 Procedure for breaking a point-to-point connection

This procedure deletes one protected connection between one connected input plug and one connected output plug. If breaking the point-to-point connection causes the output plug to become unconnected, the output plug shall stop transmitting the isochronous data flow. Figure 18 shows an implementation conforming to this procedure.

The responding application shall not reject the decrementing of the point-to-point connection counters in the OUTPUT_PLUG_CONTROL and INPUT_PLUG_CONTROL registers.

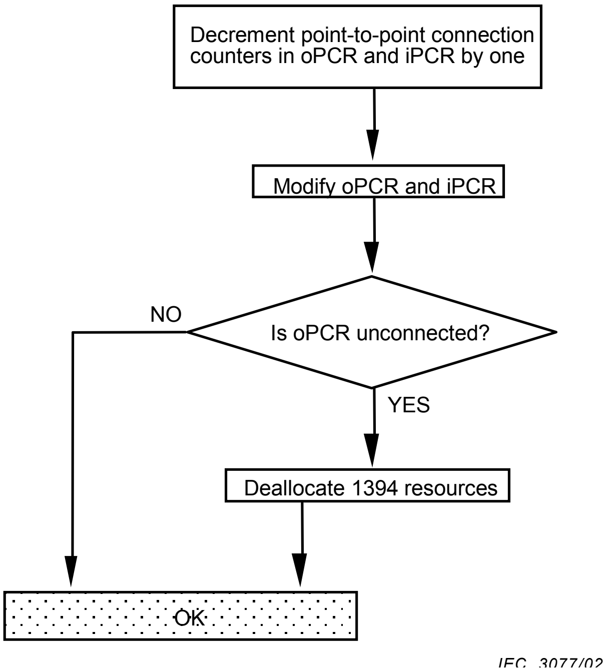


Figure 18 – Breaking a point-to-point connection

8.3 Managing broadcast-out connections

Broadcast-out connections are unprotected in the sense that a connection can be broken by any application. Consequently, the application that established the broadcast-out connection has no guarantee that the output plug will continue the transmission of the isochronous data flow. The following procedures are defined for a broadcast-out connection:

- establishing a broadcast-out connection;
- overlaying a broadcast-out connection;
- breaking a broadcast-out connection.

8.3.1 Procedure for establishing a broadcast-out connection

This procedure creates an unprotected connection between one unused channel and one unconnected output plug. Figure 19 shows an implementation conforming to this procedure.

NOTE The choice of which OUTPUT_PLUG_CONTROL register on the transmitting AV device is used does not fall within the scope of this standard. The choice of which data rate and overhead_ID are used does not fall within the scope of this standard.

The channel according to the formula in 7.5 shall be allocated. It should be noted that, if that channel is in use, the procedure fails.

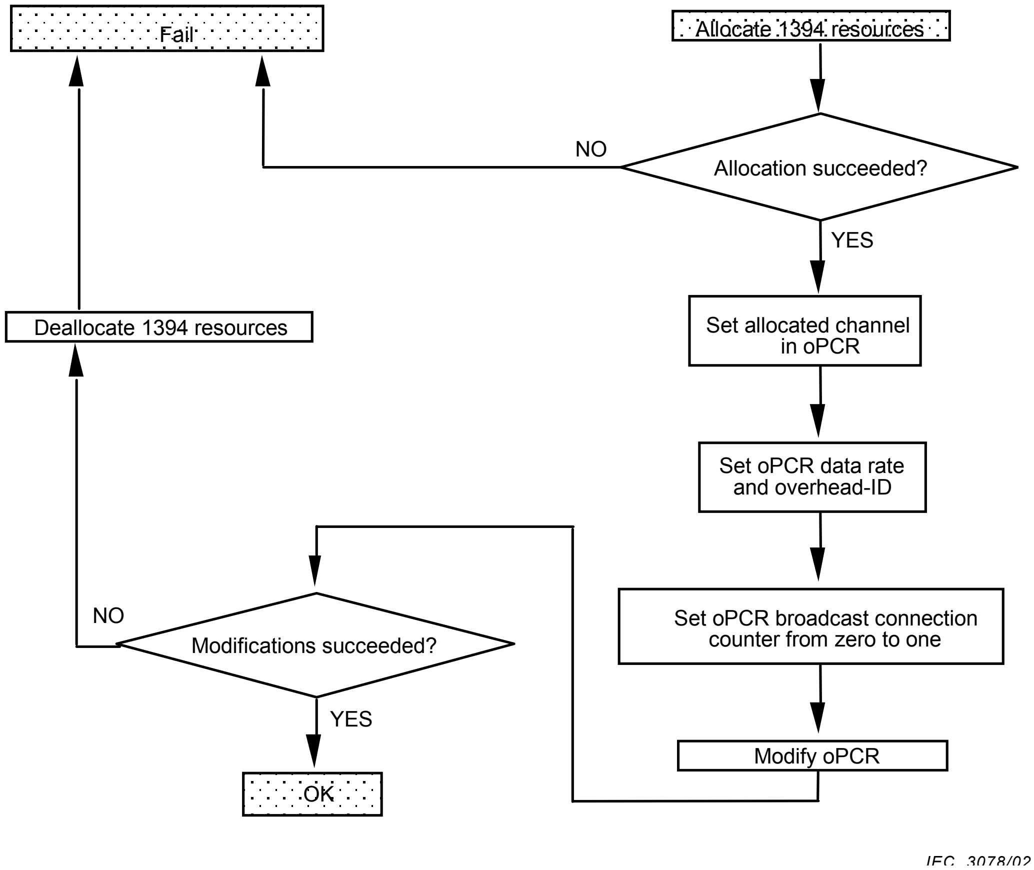


Figure 19 – Establishing a broadcast-out connection

8.3.2 Procedure for overlaying a broadcast-out connection

This procedure adds an unprotected connection between a connected output plug and the channel that this output plug uses to transmit an isochronous data flow. Figure 20 shows an implementation conforming to this procedure.

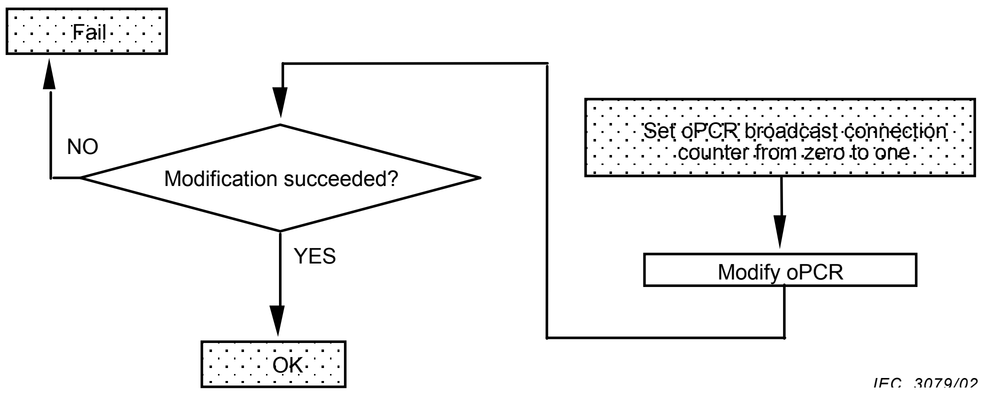


Figure 20 – Overlaying a broadcast-out connection

8.3.3 Procedure for breaking a broadcast-out connection

This procedure deletes an unprotected connection between a connected output plug and the channel that this output plug uses to transmit an isochronous data flow. If breaking the broadcast-out connection causes the output plug to become unconnected, the output plug shall stop transmitting the isochronous data flow. Figure 21 shows an implementation conforming to this procedure.

The responding application shall not reject the decrementing of the broadcast connection counter in the OUTPUT_PLUG_CONTROL register.

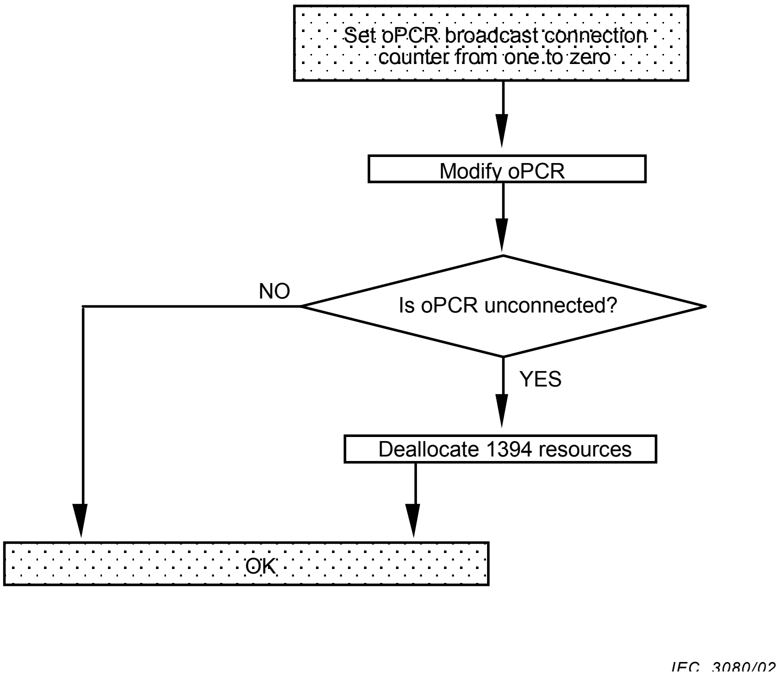


Figure 21 – Breaking a broadcast-out connection

8.4 Managing broadcast-in connections

Broadcast-in connections are unprotected in the sense that the application that established the broadcast-in connection does not know whether there is an output plug transmitting an isochronous data flow on the channel that the input plug uses to receive and, if there is, there is no guarantee that the output plug will continue the transmission.

8.4.1 Procedure for establishing a broadcast-in connection

This procedure creates an unprotected connection between one channel and one unconnected input plug. Figure 22 shows an implementation conforming to this procedure.

NOTE The choice of which INPUT_PLUG_CONTROL register on an AV device is used does not fall within the scope of this standard. The choice of which channel is used does not fall within the scope of this standard.

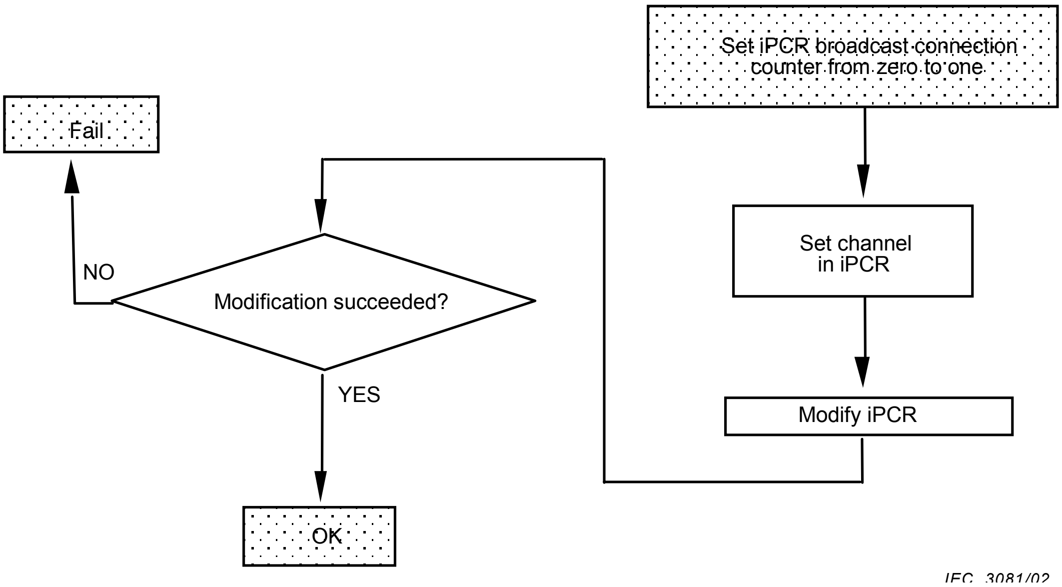


Figure 22 – Establishing a broadcast-in connection

8.4.2 Procedure for overlaying a broadcast-in connection

This procedure adds an unprotected connection between a connected input plug and the channel that this input plug uses to receive an isochronous data flow. Figure 23 shows an implementation conforming to this procedure.

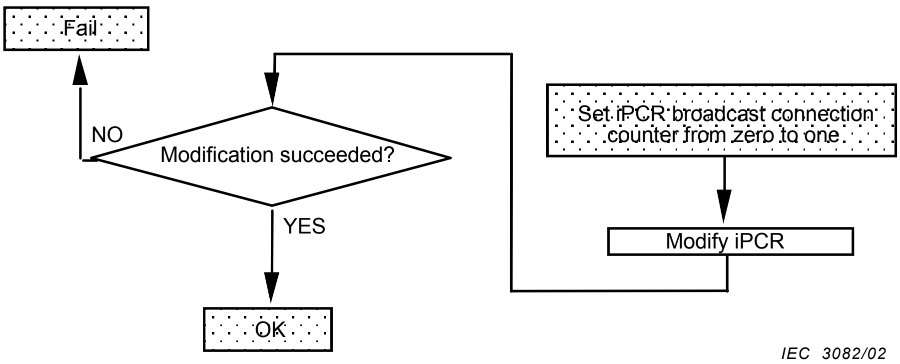


Figure 23 – Overlaying a broadcast-in connection

8.4.3 Procedure for breaking a broadcast-in connection

This procedure deletes an unprotected connection between a connected input plug and the channel that this input plug uses to receive an isochronous data flow. The input plug shall stop receiving the isochronous data flow if and only if breaking the broadcast-in connection causes the input plug to become unconnected. Figure 24 shows an implementation conforming to this procedure.

The responding application shall not reject the decrementing of the broadcast connection counter in the INPUT_PLUG_CONTROL register.

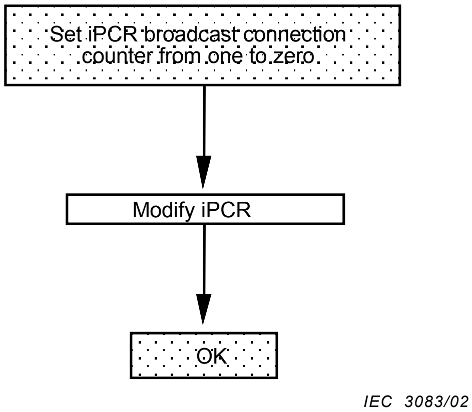


Figure 24 – Breaking a broadcast-in connection

8.5 Managing connections after a bus reset

After a bus reset, all plugs are in the unconnected state. All procedures to restore the connections that existed in a plug immediately before the bus reset shall be executed before isoch_resource_delay following the bus reset to prevent the isochronous data flows being stopped (see 7.10). In these procedures, the channel and data_rate used before the bus reset for the connection shall be used. Figure 25 shows the plug control register and isochronous data flow status after the bus reset.

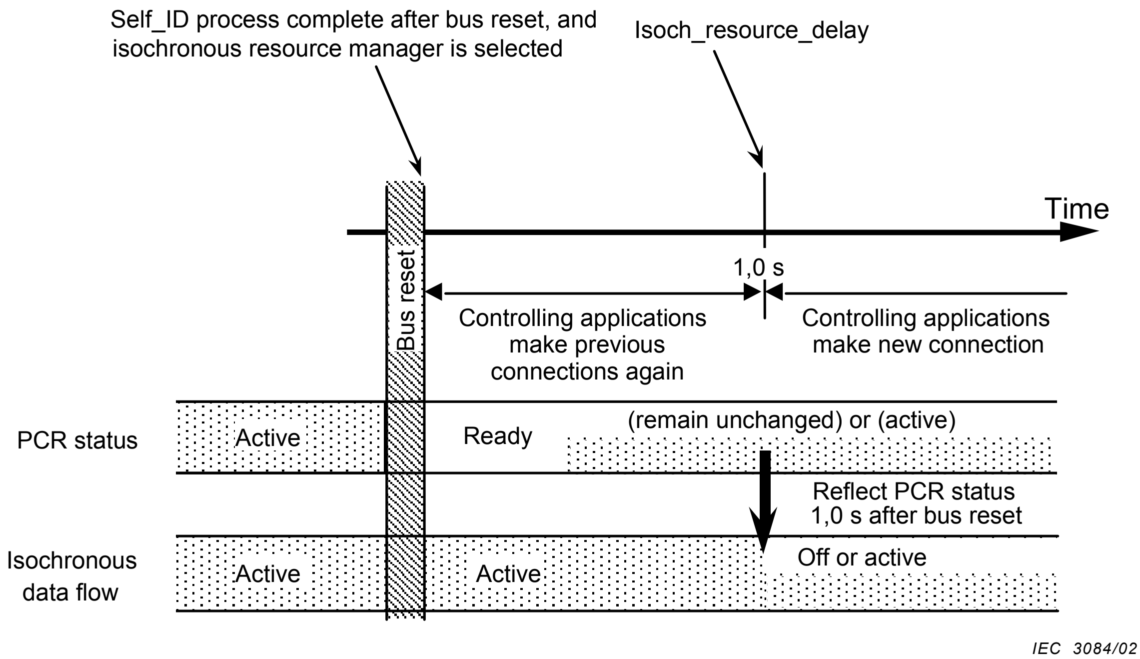


Figure 25 – Time chart of connection management and PCR activities

8.5.1 Procedure for restoring a point-to-point connection after a bus reset

Figure 26 shows an implementation conforming to the procedure to restore a point-to-point connection that it had established prior to the bus reset.

The channel and bandwidth that are to be allocated shall be calculated using the contents of the OUTPUT_PLUG_CONTROL register after the bus reset.

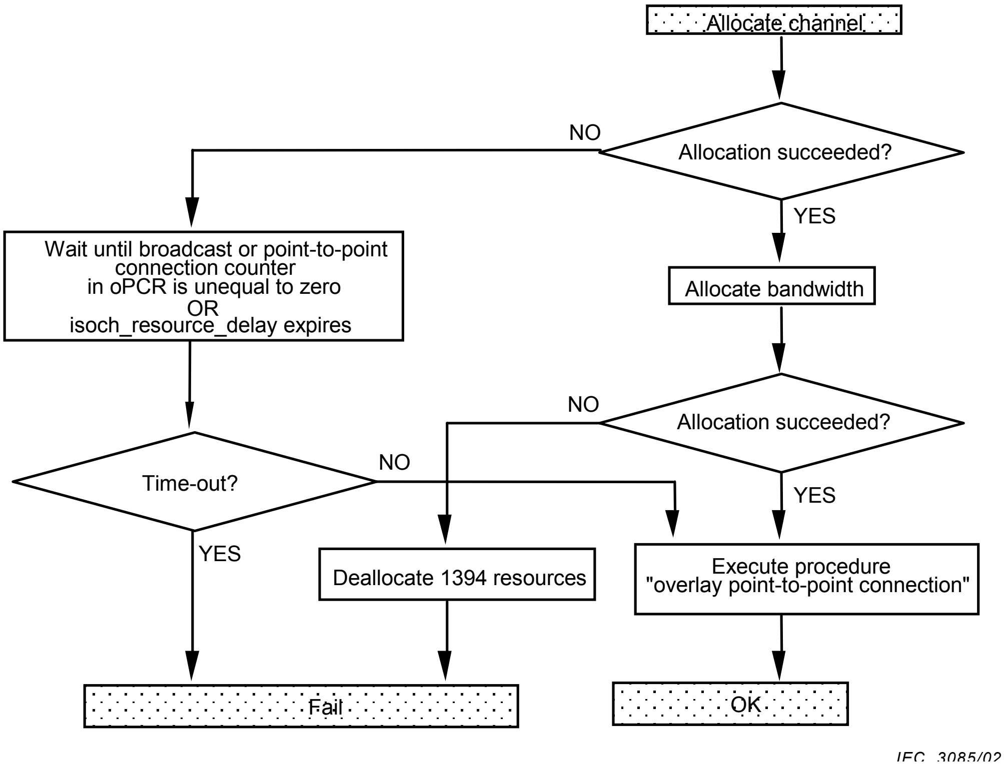


Figure 26 – Restoring a point-to-point connection

8.5.2 Procedure for restoring a broadcast-out connection after a bus reset

Figure 27 shows an implementation conforming to the procedure to restore a broadcast-out connection that it had established prior to the bus reset.

The channel and bandwidth that are to be allocated shall be calculated using the contents of the OUTPUT_PLUG_CONTROL register after the bus reset.

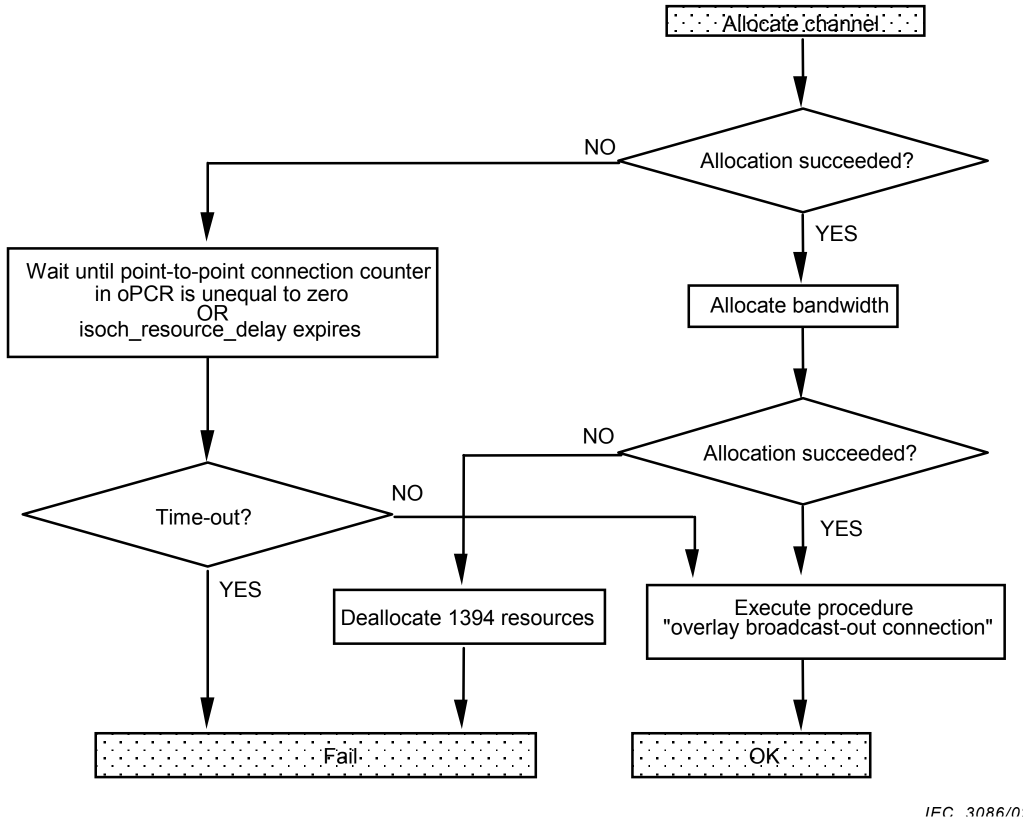


Figure 27 – Restoring a broadcast-out connection

8.5.3 Procedure for restoring a broadcast-in connection after a bus reset

Figure 28 shows an implementation conforming to the procedure to restore a broadcast-in connection that it had established prior to the bus reset.

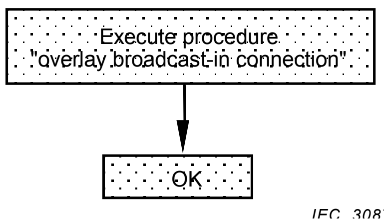


Figure 28 – Restoring a broadcast-in connection

9 Function control protocol (FCP)

9.1 Introduction

Function control protocol (FCP) is designed to control devices connected through an IEEE 1394 bus. Various command sets and command transactions are available within FCP. FCP is based on IEEE 1394 and uses asynchronous packets of IEEE 1394 for sending commands and responses. See Figure 29.

A node that controls other node(s) by FCP commands is called a controller, and a node that is controlled by FCP commands is called a target.

An FCP frame is an entity of data to be transferred from a controller to a target or vice versa. An FCP frame that is sent from a controller to a target is called a command frame, and an FCP frame that is sent from a target to a controller is called a response frame. The register that is prepared for receiving a command frame is called a command register, and the register that is prepared for receiving a response frame is called a response register.

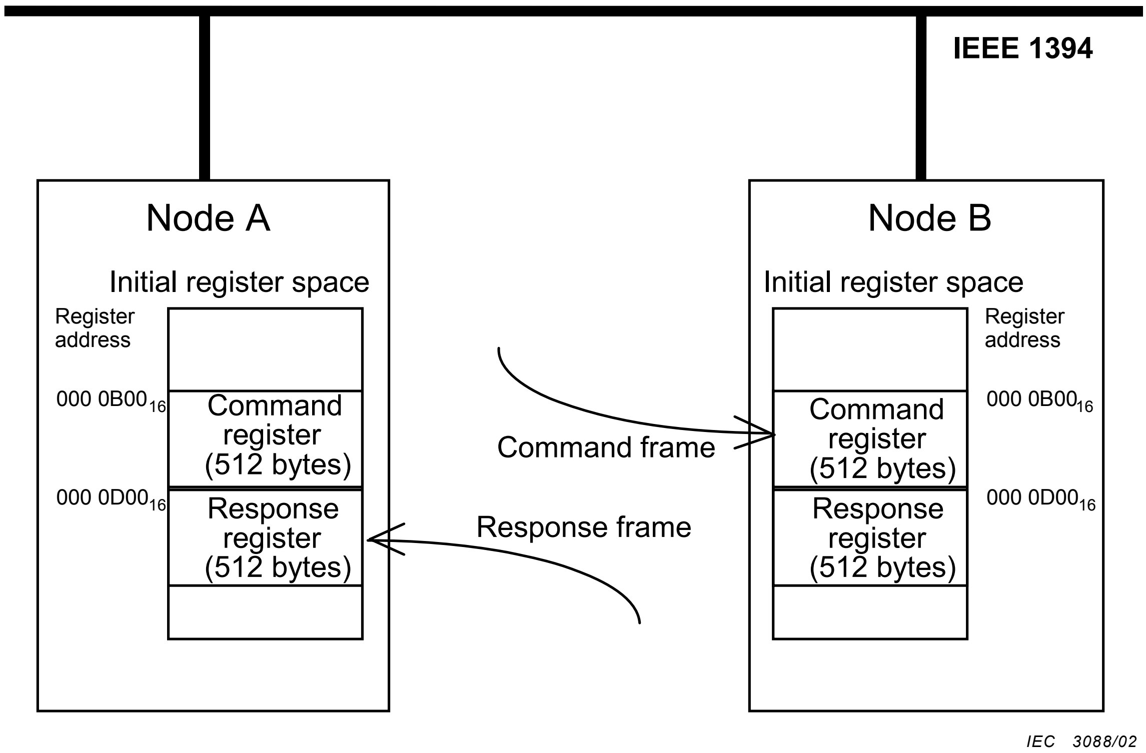


Figure 29 – Command register and response register

9.2 Asynchronous packet structure

The asynchronous packet structure used for sending an FCP frame is shown in Figures 30 and 31.

In FCP, the payloads of a write request for data block packet (refer to Figure 30) and a write request for data quadlet (refer to Figure 31) are called an FCP frame. A write request for data quadlet is used as an FCP frame only when the length of the FCP frame is exactly four bytes. FCP frames are classified as command frames and response frames. The command frame is written into a command register on a target and the response frame is written into a response register on a controller. These registers are separated and *destination_offset* addresses of these registers are specified in the FCP as below.

Base address of FCP command register (offset)	$\texttt{FFFF F000 0B00}_{16}$
Base address of FCP response register (offset)	$\texttt{FFFF F000 0D00}_{16}$

Only write requests that specify $\texttt{FFFF F000 0B00}_{16}$ or $\texttt{FFFF F000 0D00}_{16}$ as the *destination_offset* are permitted.

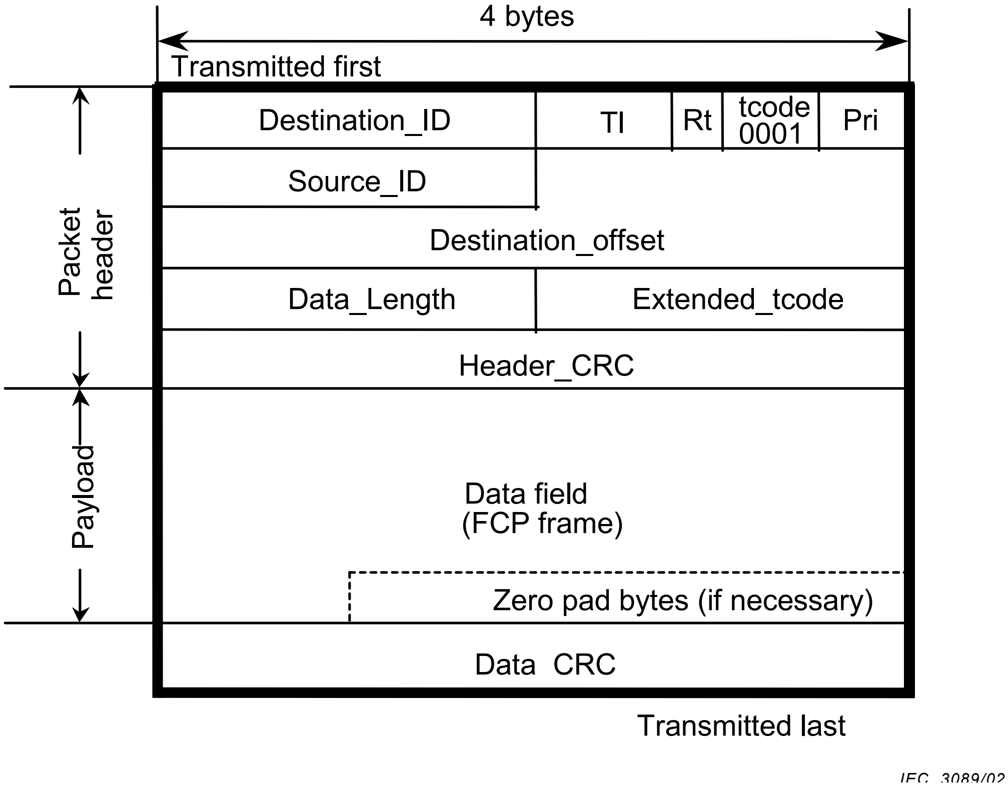


Figure 30 – Write request for data block packet of IEEE 1394

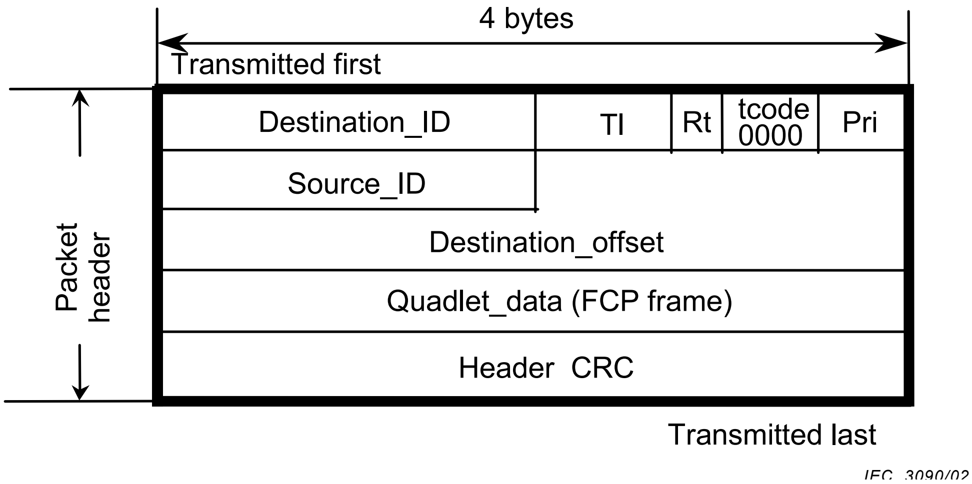


Figure 31 – Write request for data quadlet packet of IEEE 1394

9.3 FCP frame structure

The FCP frame structure is shown in Figure 32.

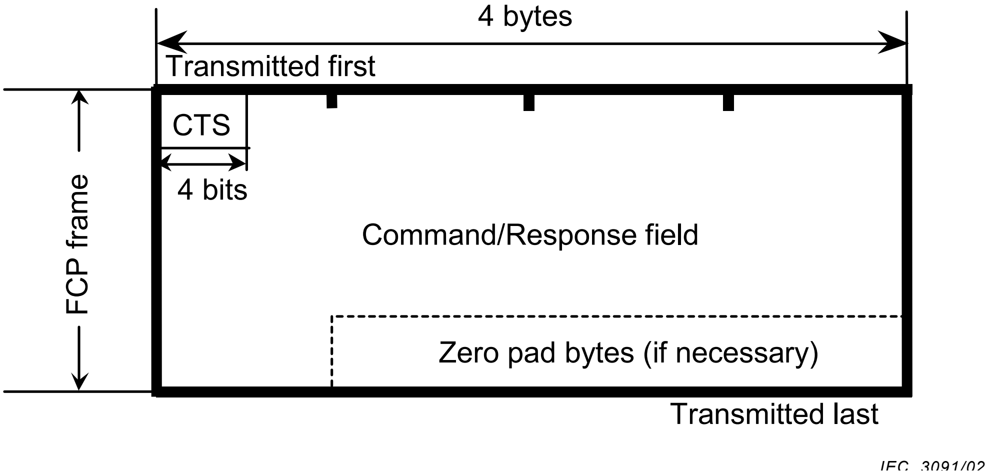


Figure 32 – FCP frame structure

Command/Transaction Set (CTS) is one component of an FCP frame. CTS specifies the command set, the structure of the command/response field and the rules of transactions used for sending commands and responses. The CTS table is shown in Table 9.

Table 9 – CTS: Command/transaction set encoding

CTS code	Command/transaction set
0000_2	AV/C
0001_2	Reserved for CAL
0010_2	Reserved for EHS
0011_2	HAVi
0100_2	Automotive
0101_2	Reserved
1110_2	Vendor unique
0110_2 to 1101_2	Reserved
1111_2	Extended CTS

9.3.1 Vendor unique command/transaction set

If the CTS code is 1110_2, it indicates that the FCP frame belongs to vendor unique CTS. An FCP frame structure that belongs to vendor unique CTS is shown in Figure 33.

Each vendor may specify a frame structure (except company_ID), a command set and rules for sending commands/responses.

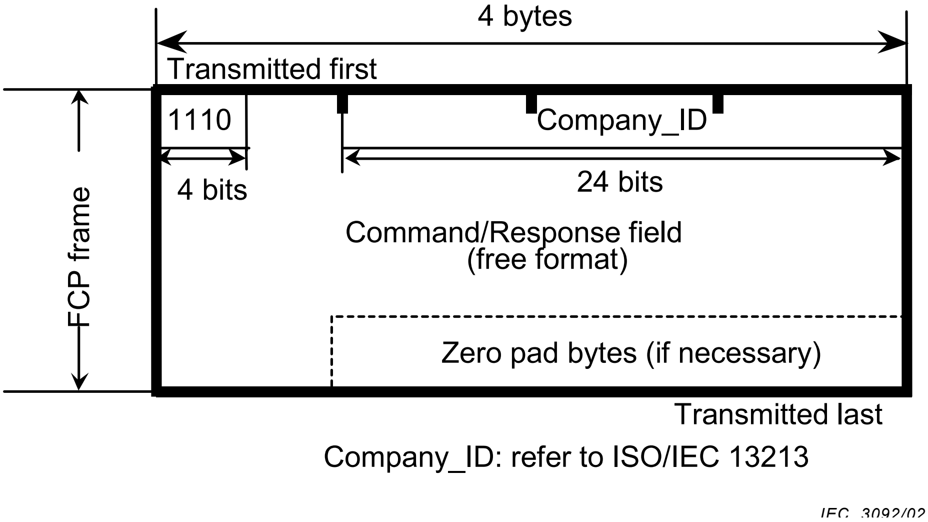


Figure 33 – Vendor unique frame format

9.3.2 Extended command/transaction set

CTS code 1111_2 is reserved for future extensions of CTS.

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

3, rue de Varembé
P.O. Box 131
CH-1211 Geneva 20
Switzerland

Tel: + 41 22 919 02 11
Fax: + 41 22 919 03 00
info@iec.ch
www.iec.ch