AV/C Digital Interface Command Set

Version 1.0
September 13, 1996

Sponsored by:
Audio/Video Working Group of the 1394 Trade Association

Approved for Release by:
1394 Trade Association Steering Committee

Abstract: This specification defines a command set for consumer audio/video equipment over 1394.0. The command set makes use of the Function Control Protocol (FCP) defined by IEC-1883, proposed standard for Digital Interface for Consumer Electronic Audio/Video Equipment, for the transport of audio/video command requests and responses. The consumer audio/video devices are implemented as a common unit architecture within 1394.0.

Keywords: Audio, Video, 1394, Digital, Interface

1394 Trade Association
c/o Skipstone, 3925 W. Braker Lane Suite 425, Austin, TX 78759 USA
http://www.1394TA.org

1394 Trade Association Specifications are developed within Working Groups of the 1394 Trade Association, a non-profit industry association devoted to the promotion of and growth of the market for IEEE 1394-compliant products. Participants in working groups serve voluntarily and without compensation from the Trade Association. Most participants represent member organizations of the 1394 Trade Association. The specifications developed within the working groups represent a consensus of the expertise represented by the participants.

Use of a 1394 Trade Association Specification is wholly voluntary. The existence of a 1394 Trade Association Specification is not meant to imply that there are not other ways to product, test, measure, purchase, market or provide other goods and services related to the scope of the 1394 Trade Association Specification. Furthermore, the viewpoint expressed at the time a specification is approved and issued is subject to change brought about through developments in the state of the art and comments received from users of the specification. Users are cautioned to check to determine that they have the latest revision of any 1394 Trade Association Specification.

Comments for revision of 1394 Trade Association Specifications are welcome from any interested party, regardless of membership affiliation with the 1394 Trade Association. Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments.

Interpretations: Occasionally, questions may arise about the meaning of specifications in relationship to specific applications. When the need for interpretations is brought to the attention of the 1394 Trade Association, the Association will initiate action to prepare appropriate responses.

Comments on specifications and requests for interpretations should be addressed to:

Editor, 1394 Trade Association
c/o Skipstone
3925 W. Braker Lane Suite 425
Austin, TX 78759
USA

Table of Contents

List of Figures

List of Tables

1 . Overview

This document specifies a command set used to control consumer electronic audio/video equipment. The command set builds upon an extensive body of standards work, underway and completed, as referenced in section 2. IEEE 1394.0, an IEEE standard, is the digital interface used to transport commands from controllers to AV devices (targets) and to return responses to the controllers. The unit architectures of these AV devices are defined within the scope of the configuration ROM and CSR architecture standardized by ISO/IEC. The commands themselves are encapsulated within a generic Function Control Protocol (FCP) developed by the HD Digital VCR Conference and now proposed as an IEC standard. Similarly, the format of the isochronous data itself has been developed by the HD Digital VCR Conference.

This specification concerns itself narrowly with the syntax and semantics of commands transmitted by controllers to AV devices and the resultant actions that occur at the AV device. The reader is strongly encouraged to read this document in conjunction with the references given below.

2 . References

IEEE Std 1394–1995, Standard for a High Performance IEEE 1394.0

IEC-1883, proposed standard for Digital Interface for Consumer Electronic Audio/Video Equipment

ISO/IEC 13123:1994, Control and Status Register (CSR) Architecture for Microcomputer Buses

HD Digital VCR Conference, Specifications of Consumer-Use Digital VCR's using 6.3 mm magnetic tape (December 1995)

3 . Definitions and abbreviations

Conformance glossary

Several keywords are used to differentiate between different levels of requirements and optionality, as follows:

expected: A keyword used to describe the behavior of the hardware or software in the design models assumed by this specification. Other hardware and software design models may also be implemented.

may: A keyword that indicates flexibility of choice with no implied preference.

shall: A keyword indicating a mandatory requirement. Designers are required to implement all such mandatory requirements to ensure interoperability with other products conforming to this specification.

should: A keyword indicating flexibility of choice with a strongly preferred alternative. Equivalent to the phrase “is recommended.”

Technical glossary

ATN: A sequential reference number recorded as part of each track of a DVCR cassette. Within the context of a single, uninterrupted recording session, these reference numbers are monotonically increasing and, in that sense, *absolute track numbers*. However, if the medium has been recorded at different times there may be gaps between different recorded areas and there is no guarantee of relationship between the absolute track numbers in one area and those in another.

AV unit: The physical instantiation of a consumer electronic device, *e.g.*, a camcorder or a VCR, within a IEEE 1394.0 node. This document describes a command set that is part of the software unit architecture for AV units.

AV subunit: an instantiation of a virtual entity that can be identified uniquely within an AV unit and offers a set of coherent functions.

AV/C: Audio/video control, as in the AV/C Digital Interface Command Set specified by this document.

byte: Eight bits of data.

CSR: A node or unit Control and Status Register, as defined by IEEE Std 1394–1995.

DVCR: Digital video cassette recorder as defined by the HD Digital VCR Conference, Specifications of Consumer-Use Digital VCR’s using 6.3 mm magnetic tape.

EUI-64: Extended Unique Identifier, 64-bits, as defined by the IEEE. The EUI-64 is a concatenation of the 24-bit company_ID obtained from the IEEE Registration Authority Committee (RAC) and a 40-bit number (typically a silicon serial number) that the vendor identified by company_ID guarantees to be unique for all of its products. The EUI-64 is also known as the node unique ID and is redundantly present in a node’s configuration ROM in both the Bus_Info_Block and the Node_Unique_Id leaf.

FCP: Function Control Protocol, as defined by IEC-1883, proposed standard for Digital Interface for Consumer Electronic Audio/Video Equipment.

IEEE: The Institute of Electrical and Electronics Engineers, Inc.

isochronous: A term that indicates the essential characteristic of a time-scale or signal, such that the time intervals between consecutive instances either have the

same duration or durations that are integral multiples of the shortest duration. In the context of IEEE 1394.0, “isochronous” is taken to mean a bounded worst-case latency for the transmission of data; physical and logical constraints that introduce jitter preclude the exact definition of “isochronous."

MIC: An acronym for memory in cassette, a feature of DVCR cassettes that provides a limited amount of nonvolatile memory that may be used for any purpose. Standard MIC formats have been specified by the HD Digital VCR Conference.

module: The smallest component of physical management, *i.e.*, a replaceable device.

nibble: Four bits of data. A byte is composed of two nibbles.

node: An addressable device attached to IEEE 1394.0 with at least the minimum set of control registers defined by IEEE Std 1394–1995.

node ID: A 16-bit number, unique within the context of an interconnected group of IEEE 1394.0es. The node ID is used to identify both the source and destination of IEEE 1394.0 asynchronous data packets. It can identify one single device within the addressable group of IEEE 1394.0es (unicast), or it can identify all devices (broadcast).

PCR: Plug Control Register, as defined by IEC-1883, proposed standard for Digital Interface for Consumer Electronic Audio/Video Equipment.

iPCR: Input plug PCR, as defined by IEC-1883.

oPCR: Output plug PCR, as defined by IEC-1883.

plug: A physical or virtual end-point of connection implemented by an AV unit or subunit that may receive or transmit isochronous or other data. Plugs may be IEEE 1394.0 plugs, accessible through the PCR's; they may be external, physical plugs on the AV unit; or they may be internal virtual plugs implemented by the AV subunits.

quadlet: Four bytes of data.

IEEE 1394.0: The physical interconnects and higher level protocols for the peer-to-peer transport of serial data, as defined by IEEE Std 1394–1995.

stream: A time-ordered set of digital data originating from one source and terminating at zero or more sinks. A stream is characterized by bounded bandwidth requirements and by synchronization points, or time stamps, within the stream data.

unit architecture: The formal specification of the format and function of the software-visible resources and behaviors of a class of units. This document, in conjunction with the references above, defines a unit architecture for the class of AV devices.

4 . Function Control Protocol (informative)

The AV/C commands and responses are transported by the Function Control Protocol (FCP) defined by IEC-1883, proposed standard for Digital Interface for Consumer Electronic Audio/Video Equipment. FCP provides a simple means to encapsulate device commands and responses within IEEE Std 1394–1995 asynchronous block write transactions. The format of an FCP frame, encapsulated within a IEEE 1394.0 block write packet, is illustrated below in Figure 4-1.

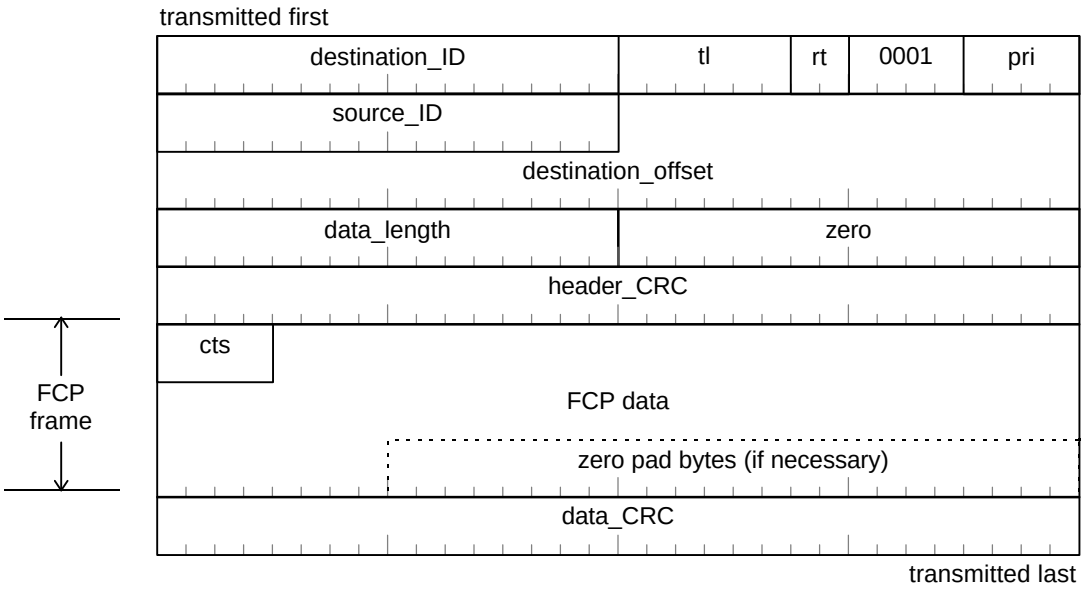

Figure 4-1 — FCP frame within a IEEE 1394.0 packet

The ***destination_ID***, ***tl***, ***rt***, ***tcode*** (write request for data block, 0001_2), ***pri***, ***source_ID***, ***data_length*** and CRC fields are as defined by IEEE Std 1394–1995.
The ***cts*** field defines the command transaction format used by the FCP frame. For the AV/C commands defined by this document, the *cts* field shall be zero.
Commands originated by a device at a IEEE 1394.0 node, the controller, are addressed to the FCP_COMMAND register, ***destination_offset*** FFFF F000 $0B00_{16}$ at the IEEE 1394.0 node that contains the device to be controlled, the target. The remotely controlled device in turn returns its response(s) to the FCP_RESPONSE register, *destination_offset* FFFF F000 $0D00_{16}$, at the controller.
The data payload of both FCP request and response packets, specified by ***data_length***, is limited to a maximum of 512 bytes.

> **NOTE:** If the size of an FCP frame is exactly four bytes, a IEEE 1394.0 quadlet write transaction shall be used to transmit the data instead of the block write packet illustrated above.

5 . AV/C frames

AV/C commands and responses are encapsulated within FCP frames, as described above, and are transmitted between the controller and target FCP_COMMAND and FCP_RESPONSE registers. The format of both the AV/C command and the AV/C response frames are similar, as described in the clauses that follow.

AV/C command frame

An AV/C command frame is up to 512 bytes of command payload with the structure shown in Figure 5-1 below.

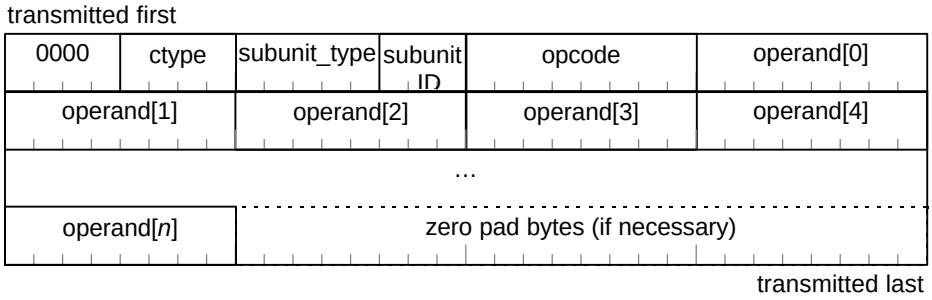

Figure 5-1 — AV/C command frame

All of the operands, up to a maximum permitted by the overall payload limit of 512 bytes, are optional and are defined by *ctype, subunit_type* and *opcode.*

> **NOTE:** If an AV/C command frame exceeds the maximum capacity of an AV unit or subunit to which it is addressed, it may be ignored.

The *subunit_type* and *subunit_ID* fields form an AV/C address which identifies the destination of the AV/C command frame and the source of the AV/C response frame.

AV/C response frame

An AV/C response frame is up to 512 bytes of response payload with the structure shown in Figure 5-2 below.

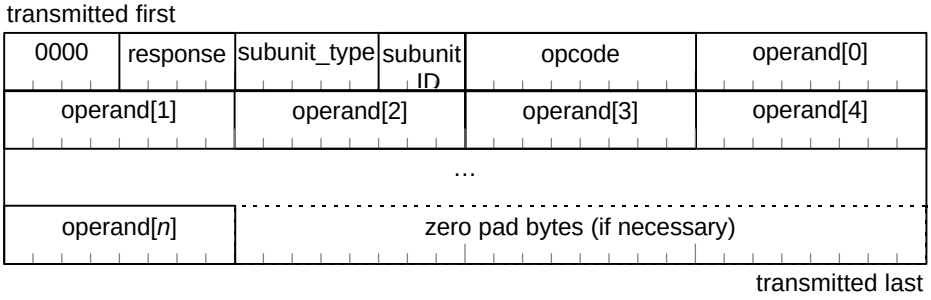

Figure 5-2 — AV/C response frame

All of the operands, up to a maximum permitted by the overall payload limit of 512 bytes, are optional and are defined by *response, subunit_type* and *opcode.*

The *subunit_type* and *subunit_ID* fields form an AV/C address which identifies the source of the responding AV/C entity and equals the destination to which the corresponding AV/C command frame was sent.

AV/C frame components

The component fields and code values for AV/C command and response frames are defined in this clause.

Except as otherwise indicated, reserved codes and fields within an AV/C frame are reserved for future specification. All reserved fields shall be set to zero by the sender of the AV/C frame. The sender shall not use reserved or invalid values for any components of an AV/C frame.

Responses to reserved or invalid codes and fields are defined in section 6.

Command type (*ctype*)

The 4-bit command type, *ctype*, defines one of four types of commands, as defined by the table below.

Value	Command type
0	CONTROL
1	STATUS
2	INQUIRY
3	NOTIFY
4 – 7	Reserved for future specification
8 – F_{16}	Reserved for response codes

Response code (*response*)

The 4-bit response code, ***response***, defines one of seven types of response, as defined by following table.

Value	Response
0 – 7	Reserved for command types
8	NOT IMPLEMENTED
9	ACCEPTED
A_{16}	REJECTED
B_{16}	IN TRANSITION
C_{16}	IMPLEMENTED / STABLE
D_{16}	CHANGED
E_{16}	Reserved for future specification
F_{16}	INTERIM

AV/C address (*subunit_type*, *subunit_ID*)

Taken together, the ***subunit_type*** and ***subunit_ID*** fields define the address of the recipient of the command or the source of the response. The values defined for *subunit_type* are defined below.

Value	Subunit type
0	Video monitor
1 – 3	Reserved for future specification
4	Video cassette recorder (VCR)
5	TV tuner
6	Reserved for future specification
7	Video camera
8 – $1B_{16}$	Reserved for future specification
$1C_{16}$	Vendor unique
$1D_{16}$. $1E_{16}$	Reserved for future specification
$1F_{16}$	Unit

Table 5-1 — Subunit type encoding

Subunit ID	Meaning
0 - 4	Instance number
5 - 6	Reserved for future specification
7	Ignore

Table 5-2 — Subunit ID encoding

An AV/C address with subunit_type value $1F_{16}$ and subunit_ID value 7 addresses the complete AV unit instead of one of its subunits. The combinations of subunit_type value $1F_{16}$ and subunit_ID values 0 through 6 are reserved.

If the subunit_type value is not equal to $1F_{16}$, the subunit_ID indicates the ordinal of the subunit as indicated by subunit_type. In this case, the subunit_ID commences at zero and is consecutively numbered up to the total instances less one, allowing a maximum of 5 instances of one subunit_type.

Operation (*opcode*)

Within the four types of AV/C commands, CONTROL, STATUS, INQUIRY and NOTIFY, the ***opcode*** field defines the operation to be performed or the status to be returned. The permissible values of *opcode* are divided into ranges valid for commands addressed to AV subunits, AV units or both, as follows.

Value	Addressing mode
0 – F_{16}	Unit and subunit commands
10_{16} – $3F_{16}$	Unit commands
40_{16} – $7F_{16}$	Subunit commands
80_{16} – $9F_{16}$	Reserved for future specification
$A0_{16}$ – BF_{16}	Unit and subunit commands
$C0_{16}$ – DF_{16}	Subunit commands
EE_{16} – FF_{16}	Reserved for future specification

Operands

The number and meaning of the ***operand[n]*** fields are determined by the *ctype*, *subunit_type* and *opcode* fields, as defined in clauses that follow.

6 . AV/C operations

AV/C commands transmitted by a controller and the associated response(s) returned by the target are called an AV/C transaction. An AV/C transaction consists of one AV/C command frame addressed to the target's FCP_COMMAND register and zero or more AV/C response frames addressed to the controller's FCP_RESPONSE register. Unless stated otherwise within individual command descriptions, it is assumed that at least one response will be returned.

The target's node_ID identifies either a specific AV unit or it identifies all AV units (broadcast). Unless stated otherwise within individual command descriptions, it is assumed that a single AV unit is addressed by the command. An example of a simple AV/C transaction, in which the target is able to complete the request before responding, is shown below in Figure 6-1.

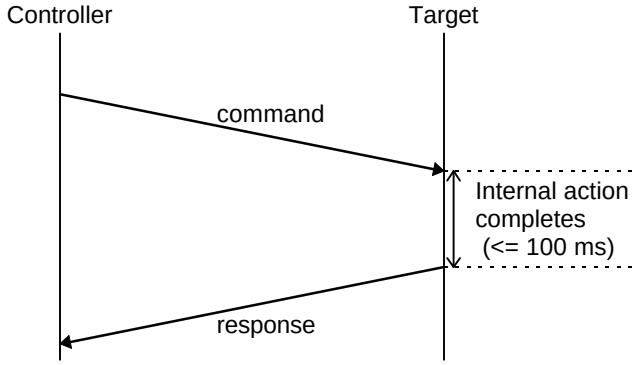

Figure 6-1 — AV/C immediate transaction

In an immediate transaction any response code, except INTERIM or CHANGED, is permitted. The transaction is complete when the target writes the AV/C response frame to the controller's FCP_RESPONSE register.

For some transactions the target may not be able to complete the request (or determine if it is possible to complete the request) within the 100 milliseconds allowed. In this case, the target shall return an initial response of INTERIM with the expectation that a final response will follow later. Figure 6-2 below illustrates an AV/C transaction with an intermediate response.

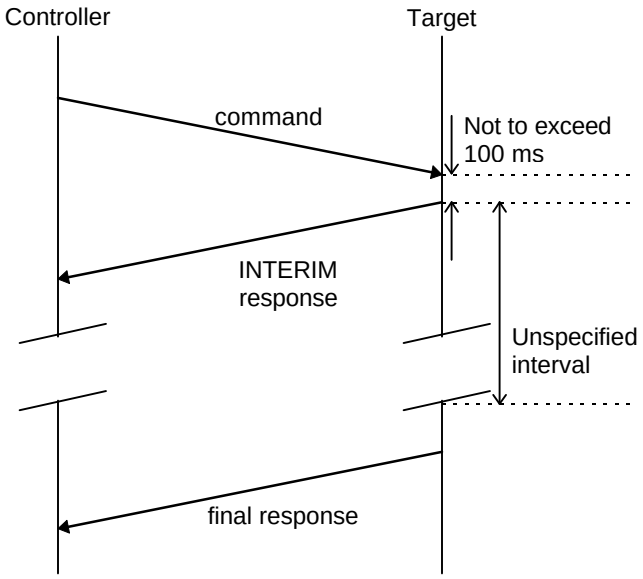

Figure 6-2 — AV/C deferred transaction

A target shall follow the following procedures when AV/C response frames are returned to the controller:

The target shall generate a response frame within 100 milliseconds of the receipt of the AV/C command frame. Targets should respond as quickly as possible.

If the AV/C command frame contains a reserved value in the *ctype* field, the target shall ignore the command and shall not generate a response frame.

If the target is already occupied with a previous command, it may ignore any AV/C command frames received. Note that the receipt of an AV/C command frame shall always be acknowledged by a target. The target ignores a command frame by a failure to return a response frame.

NOTE: A controller that does not receive a response frame for an AV/C command frame within 100 milliseconds may retry the command by resending the same command frame.

If the target is not occupied with a previous command, it shall create an AV/C response frame from the command frame by first copying all of the bytes of the command frame that precede the *opcode* and then inserting the correct response code. The remainder of the response frame, *opcode* and operands, is dependent upon the *ctype*, *subunit_type* and *opcode* of the original command.

NOTE: Response frames returned after control commands are usually identical to the original command frame except for the *response* field. A response to a status or notify command typically has different *response* and *operand* fields and, in some cases, a different *opcode* field.

If the target receives a command frame whose *subunit_type* and *subunit_ID* fields address the command to a nonexistent subunit, the target shall return a response code of NOT IMPLEMENTED.

If the target is able to initiate the requested command in less than 100 milliseconds, it shall return a response code other than INTERIM. This includes those cases where the target determines that it cannot execute the command, such as a REJECTED, response. The return of any response code other than INTERIM marks the transaction completed and the target is normally ready to accept other transactions at its FCP_COMMAND register.

If the target is unable to complete the command within 100 milliseconds, it shall promptly return an intermediate response code of INTERIM. Subsequent to an initial response of INTERIM, the target shall not send any additional

INTERIM responsesfor this command There is no time limit on command completion once an INTERIM response has been sent. The target shall ultimately send a final response when the command completes.

If the target detects a IEEE 1394.0 reset, it shall reset its state to be able to accept AV/C command frames at its FCP_COMMAND register. Any in progress transactions shall be discarded without the return of a response frame.

If a target receives an AV/C command frame using the broadcasting node_ID, and a response of NOT IMPLEMENTED would be required, no response shall be returned.

In order to correlate a response frame with an outstanding AV/C command, a controller shall examine certain fields in the response frame. The *subunit_type* and *subunit_ID* fields are never modified by the target. The *ctype* field is overwritten with the response code returned. The *opcode* and *operand[n]* fields may or may not be altered, dependent upon the command type, *subunit_type* and *opcode*. For any particular *opcode*, consult the clauses that follow for the details of the response frame returned by the target.

7 . AV/C commands

AV/C commands are variable-length strings of bytes that are embedded within a command frame and addressed to a particular AV unit or subunit. A command consists of a command type (*ctype*), a unit or subunit to which the command is addressed (*subunit_type*), an operation code (*opcode*) and one or more operands. Commands are described in the clauses that follow according to their command type, specified by *ctype* values of CONTROL, STATUS, INQUIRY or NOTIFY.

Support levels

Each AV unit or subunit may implement a subset of the AV/C command set. An unsupported command shall be rejected with a response of NOT IMPLEMENTED. Support for the different commands is characterized as mandatory, recommended, optional and vendor-dependent, as defined below:

Mandatory	The command shall be supported by any audio/video device that claims compliance with this specification and that implements the AV unit or subunit type(s) for which the command is defined. AV/C compliant devices are identified by configuration ROM entries.
Recommended	For an AV/C compliant device, the command is optional but it represents a basic functionality, *e.g.*, video and audio insert modes for a VCR subunit's RECORD command. If the device supports unit or subunit type(s) that have the functionality corresponding to the command, it is recommended that the command be implemented.
Optional	The command is optional for an AV/C compliant device.
Vendor-dependent	Support for and interpretation of the command are defined by the device vendor.

Support levels for the different commands vary according to *ctype*.

Control commands

A control command is sent by a controller to another AV device, the target, to instruct the target to perform an operation. Either the AV unit or a subunit may be the recipient of the command, as determined by the *subunit_type* and *subunit_ID* fields in the command frame. The remaining fields, *opcode* and *operand[n]*, specify the command.

Subject to the procedures described in section 6, a target that receives a control command shall return an AV/C response frame with one of the four response codes described below.

NOT IMPLEMENTED	The target does not support the control command specified by *opcode* and *operand[n]* or the command is addressed to a subunit not implemented by the target. Target state is not modified.
ACCEPTED	The target implements the control command specified by *opcode* and *operand[n]* and the target state permits execution of the command. Note that command execution may not be complete at the time a response of ACCEPTED is returned. For example, a PLAY control command sent to a VCR may be acknowledged as accepted before

the head mechanisms have engaged and the tape has started to move. The return of a response of ACCEPTED does not distinguish between a command that has completed immediately and one that is deferred but expected to complete without error.

REJECTED — The target implements the control command specified by *opcode* and *operand[n]* but the target state does not permit execution of the command. For example, a PLAY control command sent to a VCR that has no cassette inserted would be rejected..

INTERIM — If the control command specified by *opcode* and *operand[n]* is implemented but the target is unable to respond with either ACCEPTED or REJECTED within 100 milliseconds, it shall return a response frame that indicates INTERIM. Unless a subsequent bus reset causes the AV/C transaction to be aborted, the target shall ultimately return a response frame with a response code of ACCEPTED or REJECTED.

Status commands

A status command is sent by a controller to an AV device to request the device's current status. Status commands may be sent to either AV units or subunits. No target state is altered by the receipt of a status command.

NOTE: With some notable exceptions, for example the status commands that deal with a VCR's transport states, the status commands bear a family resemblance to the control commands. The same *opcode* that is used to issue a control command to a target is generally used to request corresponding status.

A target that receives a status command shall return an AV/C response frame with one of the four response codes described below:

NOT IMPLEMENTED — The target does not support the status command specified by *opcode* and *operand[n]* or the command is addressed to a subunit not implemented by the target.

REJECTED — The target implements the status command specified by *opcode* and *operand[n]* but the target state does not permit the return of status for the command.

IN TRANSITION — The target implements the status command specified by *opcode* and *operand[n]* but the target state is in transition, possibly because of an already acknowledged command or a manual operation. A subsequent status command, at an unspecified future time, may result in the return of STABLE status.

STABLE — The target implements the status command specified by *opcode* and *operand[n]* and the information requested is reported in the *opcode* and *operand[n]* values in the AV/C response frame.

NOTE: Stable information may be returned for target information that is changing because of command execution. For example, the tape position reported by a VCR may be an accurate snapshot at the time the status command was accepted, but a subsequent status command could yield a different result.

Except for the STABLE and IN TRANSITION responses, the AV/C response frame data contains the same *opcode*, operands and addressing fields as the command frame. When status information is available, both the *opcode* field and one or more of the *operand[n]* fields may be updated with the status information.

Inquiry commands

Inquiry commands may be used by a controller to determine whether or not a target AV device supports a particular control command. Except for the *ctype* field, the AV/C command frame for an inquiry command is identical to the corresponding control command.

A controller may reliably use inquiry commands to probe the capabilities of a target, since the target shall not modify any state nor initiate any command execution in response to an inquiry command.

Only two response codes, IMPLEMENTED or NOT IMPLEMENTED are permitted in the response frame returned by the target. All other fields in the response frame are exact copies of the command frame. A response of IMPLEMENTED specifies that the corresponding control command specified by *opcode* and *operand[n]* is implemented by the target AV device. An AV device implementation may validate all of the operands or it may validate only *opcode* and enough of the operands to uniquely identify the control command and determine its support level.

> **NOTE:** If a controller wishes to determine whether or not a particular status command is supported, it should issue the command. This is safe because status commands, whether or not implemented by a target, shall not cause state changes in the target.

Unlike the other command types, inquiry commands do not have a support level since they return information about the support level of the corresponding control command. However, the ability of an AV device to provide a response to an inquiry command for any *opcode* is mandatory. This insures that a controller shall always receive a response to a support level inquiry command.

The broadcasting node_ID shall not be used for inquiry commands.

Notify commands

A controller that desires to receive notification of future changes in an AV device's state may use the notify command. Responses to a notify command shall indicate the current state of the target and then, at some indeterminate time in the future, indicate the changed state of the target.

A target that receives a notify command shall not modify its state but shall generate an immediate response frame with one of the three response codes described below:

NOT IMPLEMENTED	The target does not support the notify command specified by *opcode* and *operand[n]* or the command is addressed to a subunit not implemented by the target.
REJECTED	The target implements event notification for the condition specified by *opcode* and *operand[n]* but is not able to supply the requested information.
INTERIM	The target supports the requested event notification and has accepted the notify command for any future change of state. The current state is indicated by the *opcode* and *operand[n]* data returned in the response frame. At a some future time, the target shall return an AV/C response frame with either a REJECTED or CHANGED response code.

Once a target has accepted a notify command by the return of an INTERIM response frame, the target is primed to return a subsequent response frame upon the first change in target state. The future change of target state could be the result of an operation in progress when the notify command was received or it could be the result of a control command not yet received by the target.

CHANGED	The target supports the event notification specified by *opcode* and *operand[n]* and the target state differs from the target state at the time the INTERIM response was returned. The altered target state is indicated by the *opcode* and *operand[n]* data returned in the response frame.

A typical example of the use of a notify command might involve a VCR whose cassette is being rewound. The initial response to a TRANSPORT STATE notify command is an indication of INTERIM and a “rewinding” state. When the cassette's beginning of medium is reached, the target generates a final response frame of CHANGED and a state that indicates “stopped”.

Note that notification is a one-shot operation. If the controller wishes to be notified of additional changes in a target, the controller must issue a notify command after each CHANGED response.

8 . Unit commands

Unit commands are those that are addressed to an AV device implemented as a unit architecture at a IEEE 1394.0 node. Unit commands are identified by a *subunit_type* value of $1F_{16}$ and a *subunit_ID* value of seven. Table 8-1 below summarizes the AV/C unit commands.

Opcode	Value	Support level (by *ctype*) C	S	N	Comments
CHANNEL USAGE	12_{16}	–	R	R	Report information on IEEE 1394 isochronous channel usage
CONNECT	24_{16}	O	O	R	Establish connections for unspecified streams between plugs and subunits
CONNECT AV	20_{16}	O	O	O	Establish AV connections between plugs and subunits
CONNECTIONS	22_{16}	–	O	–	Report connection status
DIGITAL INPUT	11_{16}	O	O	–	Make or break broadcast Serial
DIGITAL OUTPUT	10_{16}	O	O	–	Bus connections
DISCONNECT	25_{16}	O	–	–	Break unspecified stream connections between plugs and subunits
DISCONNECT AV	21_{16}	O	–	–	Break AV connections between plugs and subunits
INPUT PLUG SIGNAL FORMAT	19_{16}	O	R	O	Set or report signal formats for
OUTPUT PLUG SIGNAL FORMAT	18_{16}	O	R	O	IEEE 1394.0 plugs
SUBUNIT INFO	31_{16}	–	M	–	Report subunit information
UNIT INFO	30_{16}	–	M	–	Report unit information

Table 8-1 — Unit commands

A dash in the support level column indicates that the command is not defined for the *ctype* value

CONTROL, STATUS or NOTIFY, indicated.
The specific operand formats and corresponding response frame formats are described for each of the commands in the clauses that follow.

CHANNEL USAGE command

The CHANNEL USAGE status command can be used to find out which AV unit, if any, is using a particular IEEE 1394 isochronous channel.

Using a channel means that one of the AV unit's oPCRs indicates that there exists a connection which uses this channel.

For the CHANNEL USAGE status command, it is permissible to use the broadcasting node_ID.

The CHANNEL USAGE status command has the format as illustrated in Figure 8-1 below.

	msb							lsb
opcode	CHANNEL USAGE (12_{16})							
operand[0]	IEEE 1394 isochronous channel							
operand[1]	FF_{16}							
operand[2]								
operand[3]								

Figure 8-1 — CHANNEL USAGE status command format

Operand[0] denotes the isochronous channel which the target must check, to see if it is using that channel.

The CHANNEL USAGE status response has the format as illustrated in Figure 8-2 below.

	msb							lsb
opcode	CHANNEL USAGE (12_{16})							
operand[0]	IEEE 1394 isochronous channel							
operand[1]	node_ID							
operand[2]								
operand[3]	oPCR number							

Figure 8-2 — CHANNEL USAGE status response format

If in the STATUS response frame operand[1] through operand[3] are NOT all FF_{16}, it indicates that the target identified by operand[1] and operand[2] is using the channel indicated in operand[0], through the oPCR identified by operand[3]. Operand[1] contains the most significant byte of the node_ID.

If in the STATUS response frame operand[1] through operand[3] ARE all FF_{16}, it indicates that the target that returned the response is not using the channel indicated in operand[0].

In case the CHANNEL USAGE status command was broadcast (as opposed to unicast), the response obligation on this command exists only for those targets that use the channel. Because at most one target can meet this condition, at most one response frame will be returned and that response shall have operand[1] through operand[3] NOT all equal to FF_{16}.

In case the CHANNEL USAGE status command was unicast, the target shall return a response with operand[1] through operand[3] indicating whether or not the target is using the channel.

The CHANNEL USAGE command may also be used as a notify command. The notify command has the same syntax and meaning for its operands as the CHANNEL USAGE status command.

For the CHANNEL USAGE notify command, it is permissible to use the broadcasting node_ID.

In case the CHANNEL USAGE notify command was unicast and the target is not using the channel, it shall return a REJECTED response. Otherwise, it shall return an INTERIM response with operand[1] through operand[3] NOT all equal to FF_{16}. If an INTERIM response has been returned, a CHANGED response shall be returned with operand[1] through operand[3] all equal to FF_{16} once the target stops using the specified channel.

In case the CHANNEL USAGE notify command was broadcast, the response obligation on this command exists only for those targets that use the channel. Because at most one target can meet this condition, at most one INTERIM response frame will be returned with operand[1] through operand[3] NOT all equal to FF_{16}. If an INTERIM response has been returned, a CHANGED response shall be returned with operand[1] through operand[3] all equal to FF_{16} once the target stops using the specified channel.

CONNECT command

An AV *subunit* has *source* and *destination* plugs. A source plug outputs one stream from the AV subunit and a destination plug inputs one stream into the AV subunit.

An AV *unit* has *IEEE 1394.0 input* and *output* plugs to model the IEEE 1394.0 interface of the AV unit. An AV unit can have at most one IEEE 1394.0 interface and thereby at most one node ID on the IEEE 1394.0. A IEEE 1394.0 input plug inputs one stream from the IEEE 1394.0 interface into the AV unit and a IEEE 1394.0 output plug outputs one stream from the AV unit to the IEEE 1394.0 interface.

An AV unit also has *external input* and *output* plugs to model external interfaces of the AV unit other than IEEE 1394.0. An external input plug inputs one stream from an external interface into the AV unit and an external output plug outputs one stream from the AV unit to one external interface.
The CONNECT control command establishes a connection within an AV Unit between:

a source plug of an AV subunit and a destination plug of an AV subunit to carry a stream that flows inside the AV unit.

a source plug of an AV subunit and a IEEE 1394.0 or external output plug to carry a stream that flows from the AV subunit to the IEEE 1394.0 or external interface.

a IEEE 1394.0 or external input plug and a destination plug of an AV subunit to carry a stream that flows from the IEEE 1394.0 or external interface to the AV subunit.

These connections are independent from the type of data (audio, video, data, ...) inside the stream which they carry. These streams are named “unspecified streams.”

	msb							lsb
opcode	CONNECT (24_{16})							
operand[0]	$3F_{16}$						lock	perm
operand[1]	source_subunit_type					source_subunit_ID		
operand[2]	source_plug							
operand[3]	destination_subunit_type					destination_subunit_ID		
operand[4]	destination_plug							

Figure 8-3 — CONNECT control command

The subunit_type and subunit_ID fields for both the source and destination plugs have the same syntax and meaning as an AV/C address (see section 5).

The source_plug and destination_plug fields are defined by the table below:

value	source plug	destination plug
0 - $1E_{16}$	Source plug 0 - 30	Destination plug 0 - 30
$1F_{16}$ - FC_{16}	Reserved for future specification	Reserved for future specification
FD_{16}	Reserved for future specification	Multiple plugs
FE_{16}	Invalid	Invalid
FF_{16}	Any available source plug	Any available destination plug

When the stream flows from or to one of the AV unit's IEEE 1394.0 or external plugs, the *subunit_type* field shall have a value of $1F_{16}$ (AV unit) and the *subunit_ID* field shall have a value of 7. In this case, the *source_plug* and *destination_plug* fields identify either a IEEE 1394.0 or an external plug according to Table 8-2 below:

value	source plug	destination plug
0 - $1E_{16}$	IEEE 1394.0 iPCR[0] - iPCR[30]	IEEE 1394.0 oPCR[0] - oPCR[30]
$1F_{16}$ - $7E_{16}$	Reserved for future specification	Reserved for future specification
$7F_{16}$	Any available IEEE 1394.0 plug iPCR[x]	Any available IEEE 1394.0 plug oPCR[x]
80_{16} - $9E_{16}$	External input plug 0 - 30	External output plug 0 - 30
$9F_{16}$ - FC_{16}	Reserved for future specification	Reserved for future specification
FD_{16}	Reserved for future specification	Multiple plugs
FE_{16}	Invalid	Invalid
FF_{16}	Any available External input plug	Any available External output plug

Table 8-2 — IEEE 1394.0 and external plug numbers

The PLUG INFO status command may be used to determine the number of IEEE 1394.0 and external plugs of an AV unit.

Note that overlaying a connection with another connection between the same source plug and another destination plug resulting in a one-to-many flow of the same stream may or may not be allowed, depending on the capabilities of the target.

The *lock* bit pertains to the connection between the source and destination plugs as indicated in the CONNECT command. If the lock bit in the CONNECT command is set to one to establish a connection between a source and a destination plug, any subsequent CONNECT command that would result in a disruption of the stream flowing between these plugs shall return a REJECTED response. This rule shall remain valid until a subsequent DISCONNECT command has been received by the target for that source plug.

The *perm* bit is ignored in a CONNECT control command.

The CONNECT command may also be used as a status command to determine the current state of the connections within an AV unit. The CONNECT status command is used to request the identity of the source plug that is connected to a given destination plug, or the identity of the destination plug for a given source plug. The two formats for the corresponding CONNECT status commands are shown in Figure 8-4 and Figure 8-5 below, and have the same meaning as the corresponding fields of the CONNECT control command.

	msb							lsb
opcode	CONNECT (24_{16})							
operand[0]	FF_{16}							
operand[1]	source_subunit_type					source_subunit_ID		
operand[2]	source_plug							
operand[3]	FF_{16}							
operand[4]	FE_{16}							

Figure 8-4 — CONNECT status command format for a source plug

	msb							lsb
opcode	CONNECT (24_{16})							
operand[0]	FF_{16}							
operand[1]	FF_{16}							
operand[2]	FE_{16}							
operand[3]	destination_subunit_type					destination_subunit_ID		
operand[4]	destination_plug							

Figure 8-5 — CONNECT status command format for a destination plug

The CONNECT status response frame has the same format and the same meaning for all fields as the CONNECT control command except for the *perm* field.

Except for the *perm* bit, the CONNECT status response frame contains exact copies of the CONNECT operands that were used to establish the connection.

The *perm* bit in a CONNECT status response frame indicates whether a connection is permanent (value 1) or not (value 0). Permanent connections within an AV unit are connections that cannot be altered by the CONNECT control command or deleted by the DISCONNECT command, in which case a REJECTED response shall be returned.

In case there is no source plug connected to a destination plug, the *source_plug* field of the CONNECT status response frame shall indicate FE_{16} (invalid).

In case there is no destination plug connected to a source plug, the *destination_plug* field of the CONNECT status response frame shall indicate FE_{16} (invalid).

In case there are multiple destination plugs connected to a source plug, the *destination_plug* field of the CONNECT status response frame shall indicate FD_{16} (multiple plugs).

The CONNECT command may also be used as a notify command. The notify command has the same syntax as the CONNECT status command. A notification shall be returned by the target to the controller that issued the notify command in case a connection involving the plug, as indicated in the notify command, changes. These changes shall include establishing a connection to the plug, deleting a connection from the plug, and connecting the plug to another plug.

The notify responses (INTERIM and CHANGED) have the same format as the CONNECT status response frame and indicate the current status of the plug for which the notification was requested. If the plug is still connected, the plug to which it is connected shall be indicated. If the plug is no longer connected, the source or destination plug field shall be indicated as invalid (plug field value FE_{16}).

CONNECT AV command

The CONNECT AV control command is used to establish audio/video connections between subunits and plugs.

	msb							lsb
opcode	CONNECT AV (20_{16})							
operand[0]	video_source_type		audio_source_type		video_dest_type		audio_dest_type	
operand[1]	video_source							
operand[2]	audio_source							
operand[3]	video_destination							
operand[4]	audio_destination							

Figure 8-6 — CONNECT AV control command format for audio/video stream

The four fields of *operand[0], video_source_type*, *audio_source_type*, *video_dest_type* and *audio_dest_type*, encode the meaning of the four following source and destination identifying fields, as described in the table below.

Value	Source or destination type
0	Subunit
1	Ignore
2	IEEE 1394.0 or external plug
3	Reserved

If the source or destination type is zero, the corresponding source or destination operand is a subunit address, encoded as described in 5. A source or destination value of FF_{16} is a special case and indicates that the AV device may select any appropriate, available subunit.
If the source or destination type is one, the corresponding source or destination operand is ignored. This value may be used to leave existing connections unchanged or it may be used if the AV unit does not implement the connection type. For example, in a CONNECT AV control command sent to an AV unit that had only audio recording capabilities, it would be appropriate to specify a value of one for both *video_source_type* and *video_dest_type*.
If the source or destination type is two, the corresponding source or destination operand represents a IEEE 1394.0 or an external plug, as encoded by the table below.

Value	Plug
0 — $1E_{16}$	IEEE 1394.0 plug zero — 30
$1F_{16}$ — $7E_{16}$	Reserved for future specification
$7F_{16}$	Any available IEEE 1394.0 plug
80_{16} — $9E_{16}$	External plug zero — 30
$9F_{16}$ — FE_{16}	Reserved for future specification
FF_{16}	Any available external plug

NOTE: In the preceding, some of the encoded values permit the AV device to select, at its option, an available subunit, IEEE 1394.0 or external plug. The set of plugs from which the device may choose is further limited by what is appropriate. For example, a dual-deck VCR might have one deck capable of recording SD signals and another capable of recording both HD and SD signals. If a IEEE 1394.0 input plug is active and configured for HD signals, a CONNECT AV control command for an audio/video stream that specified “any available” subunit would result in the natural connection to the deck capable of recording HD signals. On the other hand, if a IEEE 1394.0 input plug is active and configured for SD

signals, an arbitrary connection could be established with either deck. In cases where more than one choice is possible, it is expected that the determination will be vendor-dependent.

In addition to its use as a control command, the CONNECT AV command may also be used as a status command to determine the current state of internal A/V connections for a unit or subunit. The form is shown in Figure 8-7 below.

	msb							lsb
opcode	CONNECT AV (20_{16})							
operand[0]	F_{16}					video_dest_type	audio_dest_type	
operand[1]	FF_{16}							
operand[2]	FF_{16}							
operand[3]	video_destination							
operand[4]	audio_destination							

Figure 8-7 — CONNECT AV status command format for audio/video stream

The fields *video_dest_type*, *audio_dest_type*, *video_destination* and *audio_destination* are used as previously described for the CONNECT AV command.

The response frame returned for the CONNECT AV status command has the same format as described in Figure 8-6.

In case there is no source plug connected to the destination plug indicated in the CONNECT AV status command, the video_source and audio_source fields shall have the value FF_{16} (invalid), and the video_source_type and audio_source_type fields shall both have the value 1 (ignore).

The CONNECT AV command may also be used as a notify command. The notify command has the same syntax as the CONNECT AV status command. A notification shall be returned by the target to the controller that issued the notify command in case a connection involving the destination as indicated in the notify command changes. These changes shall include establishing a connection to the destination, deleting a connection from the destination, and connecting the destination to another source. The notify response has the same format as the CONNECT AV response frame.

CONNECTIONS command

The CONNECTIONS status command is used to inquire the state of all connections for unspecified streams. The format of the CONNECTIONS status command is illustrated by Figure 8-8 below.

	msb							lsb
opcode	CONNECTIONS (22_{16})							
operand[0]	FF_{16}							

Figure 8-8 — CONNECTIONS status command format

The response frame returned after a CONNECTIONS status command is variable in length and depends upon the number of connections established. The response frame has the format defined by Figure 8-9 below.

	msb							lsb
opcode	CONNECTIONS (22_{16})							
operand[0]	total_connections							
operand[1]	$3F_{16}$						lock	perm
operand[2]	connection[0].source							
operand[3]								
operand[4]	connection[0].destination							
operand[5]								
…	connection[1] — connection[total_connections - 2]							
operand[*n*-4]	$3F_{16}$						lock	perm
operand[*n*-3]	connection[total_connections - 1].source							
operand[*n*-2]								
operand[*n*-1]	connection[total_connections - 1].destination							
operand[*n*]								

Figure 8-9 — CONNECTIONS response format

The *total_connections* field specifies the number of five-byte connection descriptors returned in the operands that follow. The value of *n* is determined by 5 * *total_connections*.

The format of each connection descriptor is identical to operand[1] through operand[4] of the CONNECT control command.

DIGITAL INPUT command

The DIGITAL INPUT control command permits an AV unit to establish a broadcast input connection according to its own preferences. Figure 8-10 below illustrates the format of the command.

	msb							lsb
opcode	DIGITAL INPUT (11_{16})							
operand[0]	connection_state							

Figure 8-10 — DIGITAL INPUT command format

When the DIGITAL INPUT command is issued with a *ctype* value of CONTROL, the *connection_state* field specifies whether the AV unit is expected to establish (70_{16}) or break (60_{16}) a broadcast input connection.

The DIGITAL INPUT command, with a *ctype* value of STATUS, may also be used to determine the current input broadcast connection state of the unit. In this case, *operand[0]* is set to FF_{16} when the status command is issued and is updated to the current *connection_state* when the STABLE response frame is returned.

DIGITAL OUTPUT command

The DIGITAL OUTPUT control command permits an AV unit to establish a broadcast output connection according to its own preferences. Figure 8-11 below illustrates the format of the command.

	msb							lsb
opcode	DIGITAL OUTPUT (10_{16})							
operand[0]	connection_state							

Figure 8-11 — DIGITAL OUTPUT command format

When the DIGITAL OUTPUT command is issued with a *ctype* value of CONTROL, the *connection_state* field specifies whether the AV unit is expected to establish (70_{16}) or break (60_{16}) a broadcast output connection. The AV unit shall be responsible to allocate or deallocate the necessary isochronous resources, *e.g.*, bandwidth and channel number, and to program an output PCR as appropriate.

The DIGITAL OUTPUT command, with a *ctype* value of STATUS, may also be used to determine the current output broadcast connection state of the unit. In this case, *operand[0]* is set to FF_{16} when the status command is issued and is updated to the current *connection_state* when the STABLE response frame is returned.

DISCONNECT command

The DISCONNECT control command removes a connection between a destination and a source plug for an unspecified stream as described in the CONNECT control command, even if the connection was established with the lock bit set to one. In the case where multiple connections are overlaid on the same source plug, all connections will be deleted.

The format of the DISCONNECT control command is illustrated by Figure 8-12 below.

	msb							lsb
opcode	DISCONNECT (25_{16})							
operand[0]	FF_{16}							
operand[1]	source_subunit_type					source_subunit_ID		
operand[2]	source_plug							
operand[3]	destination_subunit_type					destination_subunit_ID		
operand[4]	destination_plug							

Figure 8-12 — DISCONNECT command format

The meaning of all fields are identical to the fields as described in the CONNECT control command.

DISCONNECT AV command

The DISCONNECT AV control command is used to remove audio/video connections between subunits and plugs. The value of *operand[0]* is other than FF_{16} and the syntax is shown in Figure 8-13 below.

	msb							lsb
opcode	DISCONNECT AV (21_{16})							
operand[0]	video_source_type		audio_source_type		video_dest_type		audio_dest_type	
operand[1]	video_source							
operand[2]	audio_source							
operand[3]	video_destination							
operand[4]	audio_destination							

Figure 8-13 — DISCONNECT AV command format

The field definitions and their uses for DISCONNECT AV are identical to the field definitions given in Figure 8-6 for the CONNECT AV command.

INPUT PLUG SIGNAL FORMAT command

The INPUT PLUG SIGNAL FORMAT control command is used to configure a specified IEEE 1394.0 input plug to receive data in the designated signal format. The syntax of the INPUT PLUG SIGNAL FORMAT control command is shown in Figure 8-14 below.

	msb							lsb
opcode	INPUT PLUG SIGNAL FORMAT (19_{16})							
operand[0]	plug							
operand[1]	2			fmt				
operand[2]	(most significant byte)							
operand[3]	fdf							
operand[4]								(least significant byte)

Figure 8-14 — INPUT PLUG SIGNAL FORMAT control command format

The fields *fmt* and *fdf* are as defined in IEC-1883, proposed standard for Digital Interface for Consumer Electronic Audio/Video Equipment. Together they specify the desired signal format for the IEEE 1394.0 input plug identified by *plug*.

The INPUT PLUG SIGNAL FORMAT status command is used to inquire which signal format a specified IEEE 1394.0 input plug is configured to receive. The syntax of the INPUT PLUG SIGNAL FORMAT status command is shown in Figure 8-15 below.

	msb							lsb
opcode	INPUT PLUG SIGNAL FORMAT (19_{16})							
operand[0]	plug							
operand[1]	FF_{16}							
...								
operand[4]								

Figure 8-15 — INPUT PLUG SIGNAL FORMAT status command format

The *plug* field specifies which one of the 31 IEEE 1394.0 input plugs, zero through $1E_{16}$, is referenced.

If the status command is accepted, the response frame has the same format as the INPUT PLUG SIGNAL FORMAT control command illustrated by Figure 8-14 above. The fields *fmt* and *fdf* are as previously defined and together they specify the signal format that the IEEE 1394.0 input plug identified by *plug* is configured to receive.

The INPUT PLUG SIGNAL FORMAT command may also be used as a notify command. The notify command has the same syntax as the status command. A notification shall be returned by the target to the controller that issued the notify command in case the format of the data that the IEEE 1394.0 input plug is receiving changes. The notify response has the same format as the status response frame.

OUTPUT PLUG SIGNAL FORMAT command

The OUTPUT PLUG SIGNAL FORMAT control command is used to configure a specified IEEE 1394.0 output plug to transmit data in the designated signal format. The syntax of the OUTPUT PLUG SIGNAL FORMAT control command is shown in Figure 8-16 below.

	msb							lsb
opcode	OUTPUT PLUG SIGNAL FORMAT (18_{16})							
operand[0]	plug							
operand[1]	2	fmt						
operand[2]	(most significant byte)							
operand[3]	fdf							
operand[4]								(least significant byte)

Figure 8-16 — OUTPUT PLUG SIGNAL FORMAT control command format

The fields *fmt* and *fdf* are as defined in IEC-1883, proposed standard for Digital Interface for Consumer Electronic Audio/Video Equipment. Together they specify the desired signal format for the IEEE 1394.0 output plug identified by *plug*.

The OUTPUT PLUG SIGNAL FORMAT status command is used to inquire which signal format a specified IEEE 1394.0 output plug is configured to transmit. The format of the OUTPUT PLUG SIGNAL FORMAT command is illustrated by Figure 8-17 below.

	msb							lsb
opcode	OUTPUT PLUG SIGNAL FORMAT (18_{16})							
operand[0]	plug							
operand[1]	FF_{16}							
...								
operand[4]								

Figure 8-17 — OUTPUT PLUG SIGNAL FORMAT status command format

The *plug* field specifies which of the 31 IEEE 1394.0 output plugs, zero through $1E_{16}$, is referenced.

If the status command is accepted, the response frame has the same format as the OUTPUT PLUG SIGNAL FORMAT control command illustrated by Figure 8-16 above. The fields *fmt* and *fdf* are as previously defined and together they specify the signal format that the IEEE 1394.0 output plug identified by *plug* is configured to transmit.

The OUTPUT PLUG SIGNAL FORMAT command may also be used as a notify command. The notify command has the same syntax as the status command. A notification shall be returned by the target to the controller that issued the notify command in case the format of the data that the IEEE 1394.0 output plug is transmitting changes. The notify response has the same format as the status response frame.

SUBUNIT INFO command

The SUBUNIT INFO status command is used to obtain information about the subunit(s) of an AV unit. The format of the SUBUNIT INFO status command is illustrated by Figure 8-18 below.

	msb							lsb
opcode	SUBUNIT INFO (31_{16})							
operand[0]	0	page			0	extension_code		
operand[1]	FF_{16}							
…								
operand[4]								

Figure 8-18 — SUBUNIT INFO status command format

The *page* field value specifies which part of the subunit table is to be returned. An AV unit may implement up to 32 bytes of information in eight pages. The offset within the subunit table from which the information is returned is determined by *page* * 4.

The *extension_code* field shall have a value of seven.

If the status command is accepted, the response frame returned has the structure illustrated by Figure 8-19 below.

	msb							lsb
opcode	SUBUNIT INFO (31_{16})							
operand[0]	0	page			0	extension_code		
operand[1]	page_data							
…								
operand[4]								

Figure 8-19 — SUBUNIT INFO response format

The *page_data* returned is the four bytes from the subunit table for the page requested. The subunit table is an array of byte entries; each entry has the format defined by Figure 8-20 below.

msb							lsb
subunit_type					max_subunit_ID		

Figure 8-20 — Subunit table entry

The *subunit_type* field of each entry is as defined in Table 5-1.

The *max_subunit_ID* field is the count of subunits of *subunit_type* implemented by the AV unit, less one.

The subunit entries are not required to be in any particular order but are required to be uniquely identified by *subunit_type*. If fewer than 32 entries are present in the subunit table, they are terminated by a byte with the value FF_{16}. The value of entries past the terminating FF_{16} is indeterminate and should be ignored by any controller that requests subunit information.

UNIT INFO command

The UNIT INFO status command is used to obtain information that pertains to the AV unit as a whole (distinct from subunit information, see 8). The format of the UNIT INFO status command is illustrated by Figure 8-21 below.

	msb							lsb
opcode	UNIT INFO (30_{16})							
operand[0]	FF_{16}							
…								
operand[4]								

Figure 8-21 — UNIT INFO status command format

If the status command is accepted by the target, a response frame with the format shown by Figure 8-22 below is returned.

	msb							lsb
opcode	UNIT INFO (30_{16})							
operand[0]	07_{16}							
operand[1]	unit_type					unit		
operand[2]	(most significant byte)							
operand[3]	company_ID							
operand[4]								(least significant byte)

Figure 8-22 — UNIT INFO response format

The *unit_type* field contains a value whose meaning is identical to those defined for subunit_type in Table 5-1. Value $1C_{16}$ (vendor unique) should be returned in case none of the other values are considered to be appropriate.

The meaning of the *unit* field is not defined by this specification.

The *company_ID* field shall contain the 24-bit unique ID obtained from the IEEE Registration Authority Committee (RAC). It is expected that the value of company_ID returned by response data to the UNIT INFO status command be the same as the vendor ID in the Node Unique ID leaf in the AV unit's configuration ROM. The most significant part of the company_ID is stored in *operand[2]* and the least significant part in *operand[4]*.

9 . Common unit and subunit commands

This section defines commands that are applicable to an AV unit as well as a subunit independent of the functionality that these subunits represent indicated by their subunit_type. Table 9-1 below summarizes the common unit and subunit commands.

Opcode	Value	Support level (by *ctype*) C	S	N	Comments
POWER	$B2_{16}$	O	O	R	Control power state
RESERVE	01_{16}	O	O	R	Acquire or release exclusive control of a target
PLUG INFO	02_{16}	–	O	–	Information about IEEE 1394.0 and External plugs
VENDOR-DEPENDENT	00_{16}	V	V	V	Vendor-dependent commands

Table 9-1 — Common unit and subunit commands

In the preceding table, a dash in the support level column indicates that the command is not defined for the *ctype* value, CONTROL, STATUS or NOTIFY, indicated. The specific command formats and corresponding response frame formats are described for each of the common subunit commands in the clauses that follow.

POWER command

The POWER control command is used to control the power status of the AV unit or one of its subunits determined by the AV/C address that is contained in the AV/C frame. The format of the POWER command is illustrated by Figure 9-1 below.

	msb							lsb
opcode	POWER ($B2_{16}$)							
operand[0]	power_state							

Figure 9-1 — POWER command format

When the POWER command is issued with a *ctype* value of CONTROL, the *power_state* field specifies the desired power state of the unit. Power on is encoded as 70_{16} and power off as 60_{16}.
Setting the power status of the AV unit to on or off shall cause the power of all of its subunits to be set in the same way. Setting the power status of a subunit shall not affect the power status of the AV unit or any of the other subunits.

The POWER command with a *ctype* value of STATUS may be used to determine the current power state of the AV unit or one of its subunits. In this case, *operand[0]* is set to $7F_{16}$ when the command is issued and is updated to the current power state when the STABLE response is returned.

The POWER command may also be used as a notify command. The notify command has the same syntax as the status command. A notification shall be returned by the target to the

controller that issued the notify command in case the power state of the addressed unit or subunit changes. The notify response has the same format as the response frame.

RESERVE command

The RESERVE control command permits a controller to acquire or release exclusive control of the AV unit or one of its subunits determined by the AV/C address that is contained in the AV/C frame. The format of the command is illustrated by Figure 9-2 below.

	msb							lsb
opcode	RESERVE (01_{16})							
operand[0]	priority							
operand[1]	text							
…								
operand[12]								

Figure 9-2 — RESERVE control command format

The *priority* field shall specify the relative priority associated with the command. Zero has special meaning and indicates that no controller has reserved the AV (sub)unit. The other values, between one and $0F_{16}$, indicate that the target holds a reservation for a controller. A *priority* value of four is, by convention, the standard priority that controllers are expected to use in the absence of other reasons for choosing a higher or lower priority.

The *text* field provides for up to 12 bytes of ASCII characters. If no *text* string is present, the bytes are expected to have a value of FF_{16}.

An AV (sub)unit accepts RESERVE control commands according to the following rules:

after a power-on reset or a command reset, the AV (sub)unit is in a free state and reports a *priority* value of zero in response to any RESERVE inquiries (see the discussion of RESERVE when *ctype* has a value of STATUS, below).

a) an AV (sub)unit that is in the free state may be reserved by any controller that issues a RESERVE control command. The target shall internally record the *priority* at which the reservation is made, the *text* string that accompanies the reservation and the 16-bit node ID of the controller. An ACCEPTED response guarantees to the controller that the reservation has succeeded.

NOTE: When a priority value is accepted by an AV (sub)unit and a reservation is established, the stored value is transformed according to the following table.

Command priority	Stored priority
0 — 1	priority
02_{16} — $0E_{16}$	priority & $0E_{16}$
$0F_{16}$	priority

This has the effect of rounding most odd priorities down to a smaller even value.

b) while an AV (sub)unit holds a reservation for a controller, it shall reject any control commands other than RESERVE with a *ctype* of CONTROL that are issued by any other controller. The 16-bit node ID stored by the AV (sub)unit upon receipt of the RESERVE control command is the basis for accepting or rejecting control commands for controllers.

c) if a RESERVE control command is received from the same controller that holds the reservation, it shall be accepted. This permits the original controller to raise or lower the *priority* associated with the reservation.

d) if a RESERVE control command is received from a different controller than that which made the reservation, the AV (sub)unit shall reject the command unless the *priority* is greater than the current reservation priority. In the case where the new priority is greater than the current priority, the existing reservation is preempted and a reservation is established for the new controller according to the procedures already described in b).

e) If a RESERVE control command is addressed to the AV unit but that AV unit contains a subunit that already holds a reservation with an equal or higher priority, the RESERVE control command shall return a REJECTED response.

f) If a RESERVE control command is addressed to the AV unit and that AV unit contains no subunits that are already reserved with an equal or higher priority, then each existing reservation of a subunit shall be preempted and a reservation of the AV unit is established for the new controller according to the procedures already described in b).

g) Any control command that is addressed to a subunit within an AV unit that is reserved by a different controller than the one which issued the control command, shall be rejected.

When an AV (sub)unit detects a IEEE 1394.0 reset, it shall reset its reservation priority to zero (free) and set both the reservation node ID and the reservation text to values of all ones. Then, until the reservation has been reestablished, or until a period of ten seconds has elapsed, it shall reject all commands with a *ctype* of CONTROL except for RESERVE commands. This procedure permits the original holder of the reservation to reestablish the reservation with its reassigned node ID after the bus reset.

> **NOTE:** Controllers shall not issue RESERVE control commands within ten seconds of a bus reset unless they had established a reservation with the target AV (sub)unit prior to the bus reset. Because the node ID of the AV unit may have changed after the bus reset, a controller that wishes to reestablish (sub)unit reservations is expected to examine the unique identifier, EUI-64, in configuration ROM to locate the AV (sub)unit previously reserved.

Because of this restriction, the target can assume that a RESERVE command received within 10 seconds of a bus reset is legitimate, and shall therefore accept the reservation. Any controller may request the current reservation status of an AV (sub)unit by issuing a RESERVE command with a *ctype* field of STATUS, in the format shown in Figure 9-3.

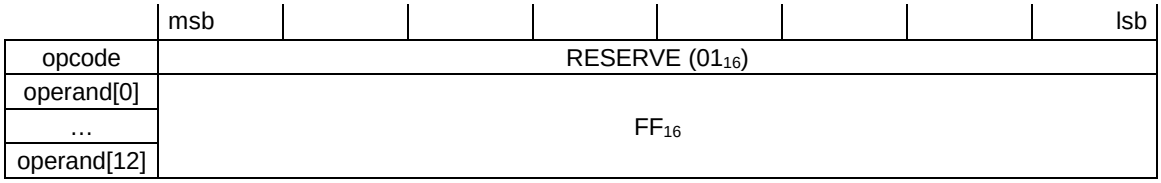

Figure 9-3 — RESERVE status command format

If a response frame is returned that indicates STABLE, *operand[0]* holds the current reservation priority and *operand[1]* through *operand[12]* hold the text string stored at the time the reservation was established. There is no way to determine the identity of the controller that holds the reservation.

Controllers that wish to be advised of a possible change of status of their own reservations, for example preemption by another controller by means of a higher priority reservation, should issue a RESERVE command in the format shown in Figure 9-3 but with a *ctype* value of NOTIFY. If a new reservation is established, the original reservation holder is notified by an AV/C response frame with CHANGED status and operand values that reflect the new reservation.

NOTE: Any new reservation results in CHANGED status, even a reservation made by the same controller that already holds a reservation. A response frame is returned to any outstanding notify command in all of these cases.

PLUG INFO command

The PLUG INFO status command is used to inquire about the number of plugs on the AV unit or one of its subunits determined by the AV/C address contained in the AV/C frame. The format of the PLUG INFO status command is illustrated by Figure 9-4 below.

	msb							lsb
opcode	PLUG INFO (02_{16})							
operand[0]	00_{16}							
operand[1]	FF_{16}							
operand[3]								
operand[4]								

Figure 9-4 — PLUG INFO status command format

If the PLUG INFO status command was addressed to an AV subunit, the format of the response frame is shown in Figure 9-5 below.

	msb							lsb
opcode	PLUG INFO (02_{16})							
operand[0]	00_{16}							
operand[1]	destination_plugs							
operand[2]	source_plugs							
operand[3]	FF_{16}							
operand[4]	FF_{16}							

Figure 9-5 — PLUG INFO status response format from an AV subunit

For the AV subunit response frame, operand[1] and operand[2] shall indicate the number of destination and source plugs of that AV subunit, and operand[3] and operand[4] shall have the value FF_{16}.

If the PLUG INFO status command was addressed to an AV unit, the response frame returned is illustrated by Figure 9-6 below.

	msb							lsb
opcode	PLUG INFO (02_{16})							
operand[0]	00_{16}							
operand[1]	Serial_Bus_input_plugs							
operand[2]	Serial_Bus_output_plugs							
operand[3]	External_input_plugs							
operand[4]	External_output_plugs							

Figure 9-6 — PLUG INFO response format from an AV unit

If the PLUG INFO status command is addressed to the AV unit, operand[1] and operand[2] shall indicate the number of IEEE 1394.0 input and output plugs, respectively, while operand[3] and operand[4] shall indicate the number of external input and output plugs, respectively.

VENDOR-DEPENDENT commands

The VENDOR-DEPENDENT command permits module vendors to specify their own set of commands and responses for AV units or subunits determined by the AV/C address that is contained in the AV/C frame. The structure of the command is illustrated by Figure 9-7 below.

	msb							lsb
opcode	VENDOR-DEPENDENT (00_{16})							
operand[0]	(most significant byte)							
operand[1]	company_ID							
operand[2]	(least significant byte)							
operand[3]	vendor_dependent_data							
…								
operand[*n*]								

Figure 9-7 — VENDOR-DEPENDENT command format

The *company_ID* field shall contain the 24-bit unique ID obtained from the IEEE Registration Authority Committee (RAC). It is expected that the value of company_ID provided in the operands of vendor-dependent commands be the same as the vendor ID in the Node Unique ID leaf in configuration ROM of the AV unit to which the command is addressed. The most significant part of the company_ID is stored in *operand[0]* and the least significant part in *operand[2]*.

The format and meaning of the *vendor_dependent_data* field are specified by the vendor identified by company_ID.

Although the behavior of vendor-dependent commands is beyond the scope of this specification, it is recommended that vendor-dependent be defined in the same four command types, CONTROL, INQUIRY, STATUS and NOTIFY, specified by the *ctype* field described in 5.

10 . VCR subunit commands

VCR subunit commands are identified by a *subunit_type* value of four and a *subunit_ID* value between zero and seven, inclusive. Table 10-1 below summarizes the VCR subunit commands.

Table 10-1 — VCR subunit commands

Opcode	Value	Support level (by *ctype*) C	S	N	Comments
ANALOG AUDIO OUTPUT MODE	70_{16}	O	O	–	Control analog audio signal
ATN	52_{16}	R	M	–	Report tape position
AUDIO MODE	71_{16}	O	O	–	Control audio signal recording mode
BACKWARD	56_{16}	R	–	–	Search for a tape position
EDIT MODE	40_{16}	O	O	–	Control editing operations prior to an anticipated playback or record command
FORWARD	55_{16}	R	–	–	Search for a tape position
INPUT SIGNAL MODE	79_{16}	O	M	–	Control input signal mode
LOAD MEDIUM	$C1_{16}$	O	–	–	Control eject, open and close
MEDIUM INFO	DA_{16}	–	R	–	Report medium information
OPEN MIC	60_{16}	*	R	–	Open or close MIC
OUTPUT SIGNAL MODE	78_{16}	O	M	–	Control output signal mode
PLAY	$C3_{16}$	*	–	–	Control the playback mode of the transport mechanism
PRESET	45_{16}	O	O	–	Establish operating parameters for the transport mechanism
READ MIC	61_{16}	R	–	–	Read data from MIC
RECORD	$C2_{16}$	*	–	–	Control the recording mode of the transport mechanism
RECORDING DATE	53_{16}	–	O	–	Report recording date
RECORDING SPEED	DB_{16}	O	O	–	Control recording speed
RECORDING TIME	54_{16}	–	O	–	Report recording time
SEARCH MODE	50_{16}	–	R	O	Report transport mechanism search mode status
TIME CODE	51_{16}	R	M	–	Search or inquire about specified medium location
TRANSPORT STATE	$D0_{16}$	–	M	O	Report current state of transport mechanism
WIND	$C4_{16}$	*	–	–	Control transport mechanism motion when not in playback or record
WRITE MIC	62_{16}	O	O	–	Store data in MIC

In the preceding table, a dash in the support level column indicates that the command is not defined for the *ctype* value, CONTROL, STATUS or NOTIFY, indicated. An asterisk in the support level column indicates that the command operands determine whether the command is mandatory (M), recommended (R) or optional (O). The specific command

formats and corresponding response frame formats are described for each of the commands in the clauses that follow.

ANALOG AUDIO OUTPUT MODE command

The ANALOG AUDIO OUTPUT MODE control command configures a VCR subunit to output an analog audio signal on its source plugs. The structure of the command is shown in Figure 10-1 below.

	msb							lsb
opcode	ANALOG AUDIO OUTPUT MODE (70_{16})							
operand[0]	analog_audio_mode							

Figure 10-1 — ANALOG AUDIO OUTPUT MODE control command format

The *analog_audio_mode* field specifies the signal format of the analog audio output, as defined by the table that follows.

Value	Analog audio mode
20_{16}	Monaural
21_{16}	Right
22_{16}	Left
23_{16}	Stereo
25_{16}	Secondary
26_{16}	Main
27_{16}	Bilingual

In addition to the use of ANALOG AUDIO OUTPUT MODE as a control command, it may also be used to determine the current audio output signal mode available at the VCR subunit's source plugs. The format used when *ctype* has a value of STATUS is illustrated by Figure 10-2.

	msb							lsb
opcode	ANALOG AUDIO OUTPUT MODE (70_{16})							
operand[0]	FF_{16}							

Figure 10-2 — ANALOG AUDIO OUTPUT MODE status command format

In the response frame returned by the VCR subunit, *operand[0]* is replaced with the current analog audio output signal format as defined above.

ATN command

The ATN command has two functions, determined by the value of *ctype* and *operand[0]*. The first requests the VCR subunit to search for a specified absolute track number on the medium while the second requests the VCR subunit to return the absolute track number value for the current medium position. The first format of the ATN command shall have a *ctype* value of CONTROL and is illustrated by Figure 10-3 below.

	msb							lsb
opcode	ATN (52_{16})							
operand[0]	20_{16}							
operand[1]	(least significant bits)							bf
operand[2]	absolute_track_number							
operand[3]	(most significant bits)							
operand[4]	FF_{16}							

Figure 10-3 — ATN control command format

The format of the *absolute_track_number* is specified by the HD Digital VCR Conference. The response frame format is identical.

The search is successful if a medium location is found such that the recorded values for both *absolute_track_number* and *bf* match the search criteria. See the discussion of *bf* below for a more detailed explanation.

The second format of the ATN command shall have a *ctype* value of STATUS and is illustrated by Figure 10-4 below.

	msb							lsb
opcode	ATN (52_{16})							
operand[0]	71_{16}							
operand[1]	FF_{16}							
…								
operand[4]								

Figure 10-4 — ATN status command format

If the VCR subunit is able to return a STABLE response to the ATN status command, the AV/C response frame has the format illustrated by Figure 10-5 below.

	msb							lsb
opcode	ATN (52_{16})							
operand[0]	71_{16}							
operand[1]	(least significant bits)							bf
operand[2]	absolute_track_number							
operand[3]	(most significant bits)							
operand[4]	FF_{16}							

Figure 10-5 — ATN response format

The *bf* bit provides discontinuity information for the absolute track numbers recorded on the medium. At the time that medium is recorded, a VCR subunit writes *bf* values of zero if either a) the present medium position does not have an absolute track number (unrecorded

medium exists prior to the current location) or b) there is at least one recorded track with a zero *bf* prior to the current location. In cases where no recording discontinuity exists between the beginning of medium and the location specified by *absolute_track_number*, the *bf* value for all tracks shall be one.

The *absolute_track_number* returned is encoded as specified by the HD Digital VCR Conference.

AUDIO MODE command

The AUDIO MODE control command specifies the recording mode format(s) for audio signals to be recorded by the VCR subunit. The AUDIO MODE control command may specify formats for up to four audio blocks, as illustrated by Figure 10-6 below.

	msb							lsb
opcode	AUDIO MODE (71_{16})							
operand[0]	audio_ctrl[3]		audio_ctrl[2]		audio_ctrl[1]		audio_ctrl[0]	
operand[1]	audio_channels[1]		sample_freq[1]		audio_channels[0]		sample_freq[0]	
operand[2]	audio_channels[3]		sample_freq[3]		audio_channels[2]		sample_freq[2]	

Figure 10-6 — AUDIO MODE control command format

> **NOTE:** A VCR subunit may record audio signals in three fundamental modes, normal, audio insert and AV insert. These modes are described in more detail in 10. An AUDIO MODE control command with operand values valid for one of these modes is not necessarily valid for the others.

An audio block is defined by whether or not it is active, the number of channels to be recorded and the sampling frequency. The HD Digital VCR Conference defines four possible audio blocks named CH1, CH2, CH3 and CH4. Each of these audio blocks is identified in the AUDIO MODE operands by an index, *n*, which may vary between zero and three, inclusive, and represents the corresponding names CH1 through CH4, respectively.

The *audio_ctrl* fields control whether or not a particular audio block is active, according to the values summarized below.

Audio control	Description
0	Record an audio signal
1	Don't record an audio signal
2	Reserved for future specification
3	Don't care

If the value of *audio_ctrl[n]* is one (or if the value of *audio_ctrl[n]* is three and the VCR subunit elects not to record an audio signal), the values for the corresponding *audio_channels[n]* and *sample_freq[n]* fields are ignored.

The *audio_channels* fields control, for enabled audio blocks, the number or format of the audio channels recorded as defined below.

Audio channels	Description
0	One audio channel
1	Two audio channels
2	20-bit audio mode
3	Don't care

NOTE: The use of 20-bit audio mode for one audio block forces the same value for all other enabled audio blocks. If different values are provided in the AV/C command frame the command may be refused with a *response* of NOT IMPLEMENTED.

The *sample_freq* fields control, for enabled audio blocks, the sampling frequency according to the following definition.

Sampling frequency	Description
0	48 kHz
1	44.1 kHz
2	32 kHz
3	Don't care

The "don't care" value in any of the AUDIO MODE operands permits the VCR subunit to select any permissible mode value for the operand. Default behavior of the VCR subunit may be obtained by means of the status command form of the AUDIO MODE command, shown in Figure 10-7 below.

	msb							lsb
opcode	AUDIO MODE (71_{16})							
operand[0]	FF_{16}							
operand[1]								
operand[2]								

Figure 10-7 — AUDIO MODE status command format

The AV/C response frame returns the current status of the VCR subunit in the operands, in the format already described in the preceding tables.

BACKWARD command

The BACKWARD control command is used to reverse the transport mechanism a specified count of units relative to the current position. Backward motion is defined as toward the beginning of medium. The format of the BACKWARD control command is illustrated by Figure 10-8 below.

	msb							lsb
opcode	BACKWARD (56_{16})							
operand[0]	measurement_unit							
operand[1]	count							
operand[2]	FF_{16}							
operand[3]								
operand[4]								

Figure 10-8 — BACKWARD control command format

The *measurement_unit* field may take on a value between zero and FE_{16}, inclusive. The HD Digital VCR Conference defines zero as a video frame and one as a video scene.
The *count* field specifies the number of the units to be moved.

EDIT MODE command

The EDIT MODE control command is used to prepare the VCR subunit for synchronized recording or playback and then to subsequently initiate synchronized recording or playback. The format of the EDIT MODE command is illustrated by Figure 10-9 below.

	msb							lsb
opcode	EDIT MODE (40_{16})							
operand[0]	subfunction							

Figure 10-9 — EDIT MODE control command format

The value of the *subfunction* operand determines the operation performed by the target, as defined by the table below.

Table 10-2 — EDIT MODE subfunctions

Subfunction	Value	Support level	Action
PREROLL & STANDBY	00_{16}	O	Search to a specified position and prepare the VCR for subsequent synchronized operations
VIDEO INSERT SYNC RECORD	21_{16}	O	Commence synchronized recording in the specified insert mode
AUDIO INSERT SYNC RECORD	22_{16}	O	
AV INSERT SYNC RECORD	23_{16}	O	
SUBCODE INSERT SYNC RECORD	24_{16}	O	
SYNC RECORD	25_{16}	O	Commence synchronized recording
SYNC PLAY	35_{16}	O	Commence synchronized playback

Before one or more EDIT MODE control commands can be issued to a target VCR subunit, the parameters used by EDIT MODE should be established by means of the PRESET control command, described in 10. EDIT MODE makes use of three parameters, PREROLL TIME, START POINT and STOP POINT.

After the appropriate parameters are established by the PRESET control command, the controller sends an EDIT MODE control command with a *subfunction* of PREROLL & STANDBY to the VCR subunit. The VCR subunit searches the medium for the time code established for the START POINT and positions the transport mechanism so that a subsequent synchronized record or playback control command may be accepted within the preroll limit.

Normally there is a significant time delay between the receipt of a PREROLL & STANDBY *subfunction* and the VCR subunit's readiness to accept a synchronized record or playback control command. The VCR subunit normally returns a response of INTERIM to the EDIT MODE control command with the PREROLL & STANDBY *subfunction* and later returns a response of ACCEPTED when the standby condition has been achieved.

Once in a standby state, the VCR subunit is ready to accept an EDIT MODE control command that specifies one of the synchronized record or playback subfunctions. This causes the VCR subunit to commence recording or playback at the START POINT after the PREROLL TIME period has elapsed. Recording or playback continues until the STOP POINT previously established.

The preceding describes the operation of the EDIT MODE command with a *ctype* of CONTROL. The same *opcode* may also be used with an *ctype* of STATUS, in which case a status response frame is returned. The formats of the EDIT MODE status command and its associated AV/C response frame are illustrated by Figure 10-10 and Figure 10-11 below.

	msb							lsb
opcode	EDIT MODE (40_{16})							
operand[0]	FF_{16}							

Figure 10-10 — EDIT MODE status command format

	msb							lsb
opcode	EDIT MODE (40_{16})							
operand[0]	edit_mode							

Figure 10-11 — EDIT MODE response format

The values returned for *edit_mode* are the same as defined for the EDIT MODE control command subfunctions in Table 10-2. If the VCR subunit is not in one of the edit modes described in Table 10-2, a value of 60_{16} is returned for *operand[0]*. A VCR subunit typically returns a response code of IN TRANSITION while positioning to the preroll point or during the preroll time period and a response code of STABLE at other times while an EDIT MODE operation is in progress.

FORWARD command

The FORWARD control command is used to advance the transport mechanism a specified count of units relative to the current position. Forward motion is defined as away from the beginning of medium. The format of the FORWARD control command is illustrated by Figure 10-12 below.

	msb							lsb
opcode	FORWARD (55_{16})							
operand[0]	measurement_unit							
operand[1]	count							
operand[2]	FF_{16}							
operand[3]								
operand[4]								

Figure 10-12 — FORWARD control command format

The *measurement_unit* field is as defined by the HD Digital VCR Conference and may take on a value between zero and FE_{16}, inclusive. Zero is defined as a video frame while one is defined as a video scene.

The *count* field specifies the number of the units to be advanced.

INPUT SIGNAL MODE command

The INPUT SIGNAL MODE control command configures a VCR subunit to accept its input data in one of the formats defined by the HD Digital VCR Conference. The structure of the INPUT SIGNAL MODE command is shown in Figure 10-13 below.

	msb							lsb
opcode	INPUT SIGNAL MODE (79_{16})							
operand[0]	signal_mode							

Figure 10-13 — INPUT SIGNAL MODE control command format

The *signal_mode* field specifies the characteristics of the input data, as defined by the table that follows.

Table 10-3 —Signal modes

Value	Signal mode
00_{16}	SD 525-60
04_{16}	SDL 525-60
08_{16}	HD 1125-60
80_{16}	SD 625-50
84_{16}	SDL 625-50
88_{16}	HD 1250-50

The INPUT SIGNAL MODE command may also be used to query which signal format the VCR subunit is currently configured to accept. In this case the *ctype* field shall be STATUS and the command format illustrated by Figure 10-14 below is used.

	msb							lsb
opcode	INPUT SIGNAL MODE (79_{16})							
operand[0]	FF_{16}							

Figure 10-14 — INPUT SIGNAL MODE status command format

The AV/C response frame returned by the VCR subunit updates *operand[0]* with one of the values described in Table 10-3 above.

LOAD MEDIUM command

The LOAD MEDIUM control command is used to control the loaded or unloaded state of medium in the transport mechanism. The format of the LOAD MEDIUM command is illustrated by Figure 10-15 below.

	msb							lsb
opcode	LOAD MEDIUM ($C1_{16}$)							
operand[0]	subfunction							

Figure 10-15 — LOAD MEDIUM control command format

The value of the *subfunction* operand determines the operation performed by the target, as defined by the table below.

subfunction	Value	Support level	Action
EJECT	60_{16}	O	Eject the cassette from the VCR
OPEN TRAY	31_{16}	O	Open the tray
CLOSE TRAY	32_{16}	O	Close the tray

MEDIUM INFO command

The MEDIUM INFO status command is used to request information about the cassette currently inserted in a VCR subunit. The format of the MEDIUM INFO status command is illustrated by Figure 10-16 below.

	msb							lsb
opcode	MEDIUM INFO (DA_{16})							
operand[0]	$7F_{16}$							
operand[1]	$7F_{16}$							

Figure 10-16 — MEDIUM INFO status command format

The information returned is formatted in a response frame, as shown by Figure 10-17 below.

	msb							lsb
opcode	MEDIUM INFO (DA_{16})							
operand[0]	medium_type							
operand[1]	write_protect							

Figure 10-17 — MEDIUM INFO response format

The first operand returned, *medium_type*, encodes information about the kind of cassette present in the VCR subunit. If the cassette type is recognized, *write_protect* encodes additional information about the cassette. Valid combinations of *medium_type* and *write_protect* are defined below.

Medium type	Write protect	Comment
DVCR standard cassette (31_{16})	30_{16}	OK to record on medium
	31_{16}	Recording inhibited
DVCR small cassette (32_{16})	30_{16}	OK to record on medium
	31_{16}	Recording inhibited
No medium present (60_{16})	$7F_{16}$	
Unknown medium ($7E_{16}$)	$7F_{16}$	Write protect status cannot be determined

OPEN MIC command

The OPEN MIC control command is used to gain access to the nonvolatile memory that may be part of a DVCR cassette. The format of the OPEN MIC control command is illustrated by Figure 10-18 below.

	msb							lsb
opcode	OPEN MIC (60_{16})							
operand[0]	subfunction							

Figure 10-18 — OPEN MIC control command format

The value of the *subfunction* operand determines the operation performed by the target, as defined by the table below.

Subfunction	Value	Support level	Action
CLOSE	00_{16}	R	Relinquish use of the MIC
READ OPEN	01_{16}	R	Open the MIC for read-only access
WRITE OPEN	03_{16}	O	Open the MIC for read or write access

A VCR subunit shall respond to OPEN MIC control commands as follows:

after a power reset, command reset or IEEE 1394.0 reset the MIC of any inserted cassette shall be in a closed state. If a cassette is inserted and is not write-protected, the VCR subunit shall accept an OPEN MIC control command from any controller.

the VCR subunit shall reject an OPEN MIC control command if no cassette is inserted, if an inserted cassette has no MIC or if an inserted cassette is write-protected and the OPEN MIC command *subfunction* specifies WRITE OPEN.

if a cassette MIC is closed or has only been opened for read access, a VCR subunit may accept any number of OPEN MIC requests with a *subfunction* of READ OPEN.

if a cassette MIC is closed or has only been opened for read access, a VCR unit may accept a single OPEN MIC control command with a *subfunction* of WRITE OPEN. This OPEN MIC operation for write access forces any existing read only opens to be closed.

if a cassette MIC is open for write access, a VCR subunit shall reject any OPEN MIC control commands except those with a *subfunction* of CLOSE sent by the controller that opened the cassette MIC for write access.

a VCR subunit shall implement a time-out period, recommended to be longer than one minute since the last accepted OPEN MIC, READ MIC or WRITE MIC control command. If this time-out period expires when a cassette MIC is open for write access, the VCR subunit shall accept an OPEN MIC control command from any controller. This forces the existing write open to be closed.

After a IEEE 1394.0 reset it is desirable for controllers that previously had opened MIC to reestablish their opened status. This is accomplished by cooperation amongst controllers. Any controller that had previously opened a MIC may issue an OPEN MIC control command immediately after a bus reset. Controllers that had not opened a MIC prior to the bus reset are expected to delay at least two seconds before issuing any OPEN MIC control commands.

NOTE: Because other control commands or manual operations may have side-effects that affect access to cassette MIC, VCR subunits are expected to reject such requests while a cassette MIC is open for either read or write access.

The OPEN MIC command may also be used with a *ctype* value of STATUS to determine the current closed or open status of a MIC. The format of this status command is illustrated by Figure 10-19 below.

	msb							lsb
opcode	OPEN MIC (60_{16})							
operand[0]	FF_{16}							

Figure 10-19 — OPEN MIC status command format

The response frame returned by the VCR subunit updates *operand[0]* to reflect the current state of the MIC or cassette, as summarized below.

Value	Meaning
00_{16}	MIC closed
01_{16}	MIC opened for read-only access by one or more controllers and able to accept additional read-only OPEN MIC commands
04_{16}	No MIC cassette inserted
08_{16}	No cassette inserted
11_{16}	MIC opened for read-only accesses and unable to accept additional read-only OPEN MIC commands
33_{16}	MIC opened for read and write accesses

OUTPUT SIGNAL MODE command

The OUTPUT SIGNAL MODE control command configures a VCR subunit to transmit its output data in one of the formats defined by the HD Digital VCR Conference. The structure of the OUTPUT SIGNAL MODE command is shown in Figure 10-20 below.

	msb							lsb
opcode	OUTPUT SIGNAL MODE (78_{16})							
operand[0]	signal_mode							

Figure 10-20 — OUTPUT SIGNAL MODE control command format

The *signal_mode* field specifies the characteristics of the output data, as defined by Table 10-3.

The OUTPUT SIGNAL MODE command may also be used to query which signal format the VCR subunit is currently configured to transmit. In this case the *ctype* field shall be STATUS and the command format illustrated by Figure 10-21 below is used.

	msb							lsb
opcode	OUTPUT SIGNAL MODE (78_{16})							
operand[0]	FF_{16}							

Figure 10-21 — OUTPUT SIGNAL MODE status command format

The AV/C response frame returned by the VCR subunit updates *operand[0]* with one of the values described in Table 10-3 above.

PLAY command

The PLAY control command is used to request the VCR transport mechanism to playback data previously recorded on the medium. The format of the PLAY control command is illustrated by Figure 10-22 below.

	msb							lsb
opcode	PLAY ($C3_{16}$)							
operand[0]	playback_mode							

Figure 10-22 — PLAY control command format

The value of *operand[0]* determines the playback mode used by the target, as defined by the following table.

Table 10-4 — Playback modes

Playback Mode	Value	Support level	Description
NEXT FRAME	30_{16}	R	Playback the next sequential frame or field
SLOWEST FORWARD	31_{16}	R	Playback at a special effect speed described in detail below
SLOW FORWARD 6	32_{16}	O	
SLOW FORWARD 5	33_{16}	O	
SLOW FORWARD 4	34_{16}	O	
SLOW FORWARD 3	35_{16}	O	
SLOW FORWARD 2	36_{16}	O	
SLOW FORWARD 1	37_{16}	O	
X1	38_{16}	O	Playback at normal speed
FAST FORWARD 1	39_{16}	O	Playback at a special effect speed described in detail below
FAST FORWARD 2	$3A_{16}$	O	
FAST FORWARD 3	$3B_{16}$	O	
FAST FORWARD 4	$3C_{16}$	O	
FAST FORWARD 5	$3D_{16}$	O	
FAST FORWARD 6	$3E_{16}$	O	
FASTEST FORWARD	$3F_{16}$	M	
PREVIOUS FRAME	40_{16}	R	Playback the previous sequential frame or field
SLOWEST REVERSE	41_{16}	R	Playback in reverse at a special effect speed described in detail below
SLOW REVERSE 6	42_{16}	O	
SLOW REVERSE 5	43_{16}	O	
SLOW REVERSE 4	44_{16}	O	
SLOW REVERSE 3	45_{16}	O	
SLOW REVERSE 2	46_{16}	O	
SLOW REVERSE 1	47_{16}	O	
X1 REVERSE	48_{16}	O	Playback at normal speed in reverse
FAST REVERSE 1	49_{16}	O	Playback in reverse at a special effect speed described in detail below
FAST REVERSE 2	$4A_{16}$	O	
FAST REVERSE 3	$4B_{16}$	O	
FAST REVERSE 4	$4C_{16}$	O	
FAST REVERSE 5	$4D_{16}$	O	
FAST REVERSE 6	$4E_{16}$	O	
FASTEST REVERSE	$4F_{16}$	M	
REVERSE	65_{16}	O	Playback at normal speed in reverse
REVERSE PAUSE	$6D_{16}$	O	Pause in reverse playback
FORWARD	75_{16}	M	Playback at normal speed
FORWARD PAUSE	$7D_{16}$	M	Pause in playback

The subunit support level for PLAY, mandatory (M), recommended (R), or optional (O), varies according to the playback mode requested.

If no cassette is loaded, the VCR subunit shall refuse the PLAY control command and return a *response* of REJECTED.

Speed variations in either a forward or reverse playback direction are collectively referred to as trick play modes. Except for FASTEST FORWARD and FASTEST REVERSE, a VCR subunit is not required to support any of the trick play modes. However, if trick play modes are supported a VCR subunit shall implement them as follows:

There are four groups of trick play modes: slow forward, fast forward, slow reverse and fast reverse. A VCR subunit may implement each group independently.

a) If a VCR subunit implements a trick play group, it shall implement the basic playback option, *i.e.*, either SLOWEST or FASTEST in the direction implemented. PLAY control commands with an *operand[0]* that specifies a SLOW *n* or FAST *n* playback mode may be rejected by the VCR subunit as NOT IMPLEMENTED. Optionally, the VCR subunit may accept all of the SLOW *n* or FAST *n* operands and interpret them as SLOWEST or FASTEST.

b) If a VCR subunit implements more than one speed within a trick play group, it shall recognize all of the SLOW *n* or FAST *n* playback modes as well as the SLOWEST or FASTEST playback mode. A VCR subunit is not required to implement all seven possible playback speeds; it is required only to map all possible playback modes within the trick play group to the speeds it does support. The actual speeds encoded by the playback modes shall be subject to one of the following restrictions, as appropriate:

SLOWEST <= SLOW 6 <= SLOW 5 <= SLOW 4 <= SLOW 3 <= SLOW 2 <= SLOW 1 <= X1

or

X1 <= FAST 1 <= FAST 2 <= FAST 3 <= FAST 4 <= FAST 5<= FAST 6 <= FASTEST

PRESET command

The PRESET control command is used to set values of internal parameters maintained by a VCR subunit. The format of the PRESET command is illustrated by Figure 10-23 below.

	msb							lsb
opcode	PRESET (45_{16})							
operand[0]	parameter_ID							
operand[1]	parameter_value							
operand[2]								
operand[3]								
operand[4]								

Figure 10-23 — PRESET control command format

The value of *parameter_ID* determines which parameter is initialized and how the four ***parameter_value*** bytes are interpreted, as described in the following table

Parameter ID		Parameter value			
Description	**Value**	**operand[1]**	**operand[2]**	**operand[3]**	**operand[4]**
START POINT	00_{16}	frame	second	minute	hour
STOP POINT	01_{16}				
PREROLL TIME	02_{16}			FF_{16}	FF_{16}

The time values, *frame*, *second*, *minute* and *hour*, are all encoded in binary coded decimal (BCD) format where each nibble of a byte represents a decimal digit. The *minute* and *hour* values are not used for PREROLL TIME.

If a VCR subunit implements the PRESET control command but does not support a particular *parameter_value* for one of the parameters, it may generate a REJECTED response to the command.

NOTE: One example of this is given by VCR subunits that implement fixed preroll times. If a PRESET control command to set the preroll time specifies frame and second values different from the fixed preroll time, the VCR subunit shall reject the command.

The PRESET command may also be used with a *ctype* value of STATUS. This status command, whose format is shown by Figure 10-24 below, reports the current value of the specified parameter in the response frame returned.

	msb							lsb
opcode	PRESET (45_{16})							
operand[0]	parameter_ID							
operand[1]	FF_{16}							
…								
operand[4]								

Figure 10-24 — PRESET status command format

The current value of the parameter is reported by the *operand[n]* bytes returned in the response frame.

READ MIC command

The READ MIC control command is used to access variable-length data stored in cassette nonvolatile memory (MIC). A controller is expected to have issued an OPEN MIC control command to the target subunit before performing any MIC read or write operations. The format of the READ MIC control command is illustrated by Figure 10-25 below.

	msb							lsb
opcode	READ MIC (61_{16})							
operand[0]	data_length							
operand[1]	(most significant byte) MIC_address							
operand[2]								(least significant byte)

Figure 10-25 — READ MIC control command format

The *data_length* field specifies the number of bytes to be read from the MIC.

The *MIC_address* field specifies the starting offset of the data within the MIC.

If an ACCEPTED response frame is returned by the target after a READ MIC control command, the response data consists of additional operands, *operand[3]* through *operand[data_length + 2]*, that contain the data bytes requested.

If the VCR subunit is not be able to return the number of bytes indicated by *data_length* in a single operation, it shall return the maximum quantity of data it is able and adjust the *data_length* field in the response frame accordingly.

If the combination of *MIC_address* and *data_length* address unimplemented memory, the VCR subunit may return the addressable data and adjust the *data_length* field in the response frame accordingly.

RECORD command

The RECORD control command is used to request the VCR transport mechanism to record signal(s) on the medium. The format of the RECORD control command is illustrated by Figure 10-26 below.

	msb							lsb
opcode	RECORD ($C2_{16}$)							
operand[0]	recording_mode							

Figure 10-26 — RECORD control command format

The value of *operand[0]* determines the recording mode used by the target, as defined by the table below.

Table 10-5 — Recording modes

Recording Mode	Value	Support level	Description
VIDEO INSERT	31_{16}	R	Replace the specified type of signal with a new signal but leave the other signals(s) on the medium intact
AUDIO INSERT	32_{16}	R	
AV INSERT	33_{16}	R	
SUBCODE INSERT	34_{16}	O	
VIDEO INSERT PAUSE	41_{16}	R	Pause recording signal(s) on the medium and establish the recording mode indicated
AUDIO INSERT PAUSE	42_{16}	R	
AV INSERT PAUSE	43_{16}	R	
SUBCODE INSERT PAUSE	44_{16}	O	
RECORD	75_{16}	M	Overwrite all signal(s) on the medium
RECORD PAUSE	$7D_{16}$	M	Pause while recording all signal(s)

The subunit support level for RECORD, mandatory (M), recommended (R), or optional (O), varies according to the recording mode requested.

> **NOTE:** If a VCR subunit does not have record capabilities, none of the recording modes for the RECORD control command are mandatory.

If no cassette is loaded or a write-protected cassette is loaded, the VCR subunit shall refuse the RECORD control command and return a *response* of REJECTED.

RECORDING DATE command

The RECORDING DATE status command requests the VCR subunit to return the recording date of the inserted cassette. The format of the RECORDING DATE status command is illustrated by Figure 10-27 below.

	msb							lsb
opcode	RECORDING DATE (53_{16})							
operand[0]	71_{16}							
operand[1]	FF_{16}							
…								
operand[4]								

Figure 10-27 — RECORDING DATE status command format

If the VCR subunit is able to return a STABLE response to the RECORDING DATE status command, the AV/C response frame has the format illustrated by Figure 10-28 below.

	msb							lsb
opcode	RECORDING DATE (53_{16})							
operand[0]	71_{16}							
operand[1]	ds	tm	time_zone					
operand[2]	3		day					
operand[3]	weekday			month				
operand[4]	year							

Figure 10-28 — RECORDING DATE response format

The *ds* bit indicates whether or not daylight saving time was in force at the time of recording. A value of zero indicates daylight saving time while a value of one indicates standard time.

The *tm* bit indicates whether or not the time zone is offset by 30 minutes from Greenwich Meridian Time (GMT). A value of zero indicates that local time equals GMT plus *time_zone* plus 30 minutes while a value of one indicates that no offset is necessary.

The *time_zone* field specifies the hours offset from GMT. The *time_zone* field is encoded in binary coded decimal (BCD) format where the most significant two bits and the least significant nibble each represent a decimal digit. The *time_zone* field shall have a value between zero and 23, inclusive, or a value of $3F_{16}$ (which indicates that no time zone information is available).

The *day* field specifies the day of the month. The *day* field is encoded in binary coded decimal (BCD) format where the most significant two bits and the least significant nibble each represent a decimal digit. The *day* field shall have a value between one and 31, inclusive, or a value of $3F_{16}$ (which indicates that no day of the month information is available).

The *weekday* field specifies the day of the week and is encoded according to the following table.

Value	Day of the week
0	Sunday
1	Monday
2	Tuesday
3	Wednesday
4	Thursday
5	Friday
6	Saturday
7	No information available

The *month* field is encoded in binary coded decimal (BCD) format where the most significant bit and the least significant nibble each represent a decimal digit. The *month* field shall have a value between one and twelve, inclusive, that represents the months January through December, respectively, or a value of $1F_{16}$ (which indicates that no month information is available).

The *year* field specifies the least significant two digits of the year, *Anno Domini* (AD). The *year* field is encoded in binary coded decimal (BCD) format where the each nibble represents a decimal digit. The *year* field shall have a value between zero and 99, inclusive, or a value of FF_{16} (which indicates that no year information is available).

RECORDING SPEED command

The RECORDING SPEED command is used to set or query the recording speed for the VCR subunit's transport mechanism. The format of the RECORDING SPEED control command is illustrated by Figure 10-29 below.

	msb							lsb
opcode	RECORDING SPEED (DB_{16})							
operand[0]	recording_speed							

Figure 10-29 — RECORDING SPEED control command format

The value of *operand[0]* determines the recording speed to be used by the VCR subunit. The *recording_speed* operand shall have a value between zero and FE_{16}, inclusive. A *recording_speed* value of $6F_{16}$ represents a standard recording speed; values less than $6F_{16}$ are considered lower recording speeds while values greater than $6F_{16}$ are considered higher speeds.

Two values for *recording_speed* are specified by the HD Digital VCR Conference, as defined by the table below.

Recording Mode	Value	Comment
Speed 32	20_{16}	6.67 µm track pitch
Standard speed	$6F_{16}$	10 µm track pitch

In addition to a *ctype* value of CONTROL, the RECORDING SPEED command is also valid with a *ctype* of STATUS. This form of the status command has the format shown below by Figure 10-30.

	msb							lsb
opcode	RECORDING SPEED (DB_{16})							
operand[0]	$7F_{16}$							

Figure 10-30 — RECORDING SPEED status command format

In the response frame generated by the VCR subunit, the current recording speed is returned as *operand[0]*.

RECORDING TIME command

The RECORDING TIME status command requests the VCR subunit to return the recording time of the inserted cassette. The format of the RECORDING TIME status command is illustrated by Figure 10-31 below.

	msb							lsb
opcode	RECORDING TIME (54_{16})							
operand[0]	71_{16}							
operand[1]	FF_{16}							
…								
operand[4]								

Figure 10-31 — RECORDING TIME status command format

If the VCR subunit is able to return a STABLE response to the RECORDING TIME status command, the AV/C response frame has the format illustrated by Figure 10-32 below.

	msb							lsb
opcode	RECORDING TIME (53_{16})							
operand[0]	71_{16}							
operand[1]	s2	s1	frames					
operand[2]	s3	seconds						
operand[3]	s4	minutes						
operand[4]	s6	s5	hours					

Figure 10-32 — RECORDING TIME response format

The *s1*, *s2*, *s3*, *s4* , *s5* and *s6* fields collectively encode SMPTE/EBU time code information. When this information is not available, all of these fields shall be set to one. Otherwise, only one of the fields shall be set to one and represents information encoded by the table below.

SMPTE/EBU data	**s1**	**s2**	**s3**	**s4**	**s5**	**s6**
Vertical interval time code (VITC)	14	15	35	55	74	75
Linear time code (LTC)	10	11	27	43	58	59

Each of the *frames*, *seconds*, *minutes* and *hours* fields is encoded in binary coded decimal (BCD) format where the least significant nibble and the remaining most significant bits are each a field that represents a decimal digit.

SEARCH MODE command

The SEARCH MODE status command is used to query the current search mode and search destination, if any, of a VCR subunit. The format of the SEARCH MODE status command is illustrated by Figure 10-33 below.

	msb							lsb
opcode	SEARCH MODE (50_{16})							
operand[0]	FF_{16}							
…								
operand[4]								

Figure 10-33 — SEARCH MODE status command format

The response frame returned after a SEARCH MODE status command differs from many others in that the *opcode* is updated with a result value as well as all five operands. The possible combinations for the return of *opcode* and the resultant operands are defined in the following table.

Opcode	Value	operand[0]	operand[1]	operand[2]	operand[3]	operand[4]
SEARCH MODE	50_{16}	OTHER MODE (60_{16})	FF_{16}	FF_{16}	FF_{16}	FF_{16}
TIME CODE	51_{16}	SEARCHING (20_{16})	frame	second	minute	hour
ATN	52_{16}	SEARCHING (20_{16})	absolute_track_number			FF_{16}
FORWARD	55_{16}	measurement_unit	count	FF_{16}	FF_{16}	FF_{16}
BACKWARD	56_{16}	measurement_unit	count	FF_{16}	FF_{16}	FF_{16}

If the VCR subunit's transport mechanism is not active in either a TIME CODE, ATN, FORWARD or BACKWARD search, the same *opcode*, 50_{16}, is returned and the first operand indicates that the VCR subunit is not in one of these search modes. Otherwise, the *opcode* is updated with the *opcode* used to commence the search and the operands reflect the intended destination of the search. The TIME CODE values, *frame*, *second*, *minute* and *hour*, are all encoded in binary coded decimal (BCD) format where each nibble of a byte represents a decimal digit. The *absolute_track_number* returned is encoded in the same fashion as described in 10. The *measurement_unit* and *count* values are as described in sections 10 and 10.

The SEARCH MODE command may also be used as a notify command. The SEARCH MODE notification service is useful to controllers who are monitoring a subunit. The actions for which SEARCH MODE provides status information can take a very long time to complete. Upon discovering that a subunit is engaged in one of these long operations, a controller can issue a SEARCH MODE notify command, so that it will be notified when the operation terminates. Note that the operation may or may not terminate successfully.

The target shall send a CHANGED response whenever the status as maintained by the SEARCH MODE command changes. The CHANGED response shall indicate the latest SEARCH status or OTHER MODE if the target has terminated the SEARCH MODE. The termination may be because the destination was found, or because it was not found and the end of the media was reached, or because it was interrupted by a higher priority event such as the user pressing the STOP button.

TIME CODE command

The TIME CODE command has two functions, determined by the value of *ctype* and *operand[0]*. The first requests the VCR subunit to search for a specified time code on the medium while the second requests the VCR subunit to return the time code value for the current medium position. The first format of the TIME CODE command shall have a *ctype* value of CONTROL and is illustrated by Figure 10-34 below.

	msb							lsb
opcode	TIME CODE (51_{16})							
operand[0]	20_{16}							
operand[1]	frame							
operand[2]	second							
operand[3]	minute							
operand[4]	hour							

Figure 10-34 — TIME CODE control command format

The values, *frame*, *second*, *minute* and *hour*, are all encoded in binary coded decimal (BCD) format where each nibble of a byte represents a decimal digit. The response frame format is identical.

The second format of the TIME CODE command shall have a *ctype* value of STATUS and is illustrated by Figure 10-35 below.

	msb							lsb
opcode	TIME CODE (51_{16})							
operand[0]	71_{16}							
operand[1]	FF_{16}							
…								
operand[4]								

Figure 10-35 — TIME CODE status command format

If the VCR subunit is able to return a STABLE response to the TIME CODE status command, the AV/C response frame has the format illustrated by Figure 10-36 below.

	msb							lsb
opcode	TIME CODE (51_{16})							
operand[0]	71_{16}							
operand[1]	frame							
operand[2]	second							
operand[3]	minute							
operand[4]	hour							

Figure 10-36 — TIME CODE response format

The values, *frame*, *second*, *minute* and *hour*, are all encoded in binary coded decimal (BCD) format where each nibble of a byte represents a decimal digit.

TRANSPORT STATE command

The TRANSPORT STATE status or notify command is used to inquire as to the current state of the medium in the transport mechanism. The format of the TRANSPORT STATE command is illustrated by Figure 10-37 below.

	msb							lsb
opcode	TRANSPORT STATE ($D0_{16}$)							
operand[0]	$7F_{16}$							

Figure 10-37 — TRANSPORT STATE command format

The TRANSPORT STATE command shall be used only with a *ctype* of STATUS or NOTIFY. The information returned is formatted in a response frame, as shown by Figure 10-38 below.

	msb							lsb
opcode	transport_mode							
operand[0]	transport_state							

Figure 10-38 — TRANSPORT STATE response format

In the response frame returned after a TRANSPORT STATE status or notify command, the ***transport_mode*** indicates the operating mode of the transport mechanism. The transport state is represented by previously defined opcodes, LOAD MEDIUM, RECORD, PLAY and WIND. Within each category of *transport_mode*, the value returned for *operand[0]* represents the transport state, as defined by Table 10-6 below.

Table 10-6 — Transport state values

Transport mode	Transport state	Value	Comment
LOAD MEDIUM ($C1_{16}$)	EJECT	60_{16}	No medium is present
RECORD ($C2_{16}$)	As defined by Table 10-5		Any of the values defined for the RECORD command may be returned to indicate the corresponding transport state
	UNSPECIFIED INSERT	30_{16}	One of the insert states, 31_{16} - 34_{16}
	UNSPECIFIED INSERT PAUSE	40_{16}	One of the paused insert states, 41_{16} - 44_{16}
PLAY ($C3_{16}$)	As defined by Table 10-4		Any of the values defined for the PLAY command, except 30_{16} and 40_{16}, may be returned to indicate the corresponding transport state
	UNSPECIFIED SLOW FORWARD	31_{16}	One of the SLOW FORWARD trick play states, 31_{16} - 37_{16}
	UNSPECIFIED FAST FORWARD	$3F_{16}$	One of the FAST FORWARD trick play states, 39_{16} - $3F_{16}$
	UNSPECIFIED SLOW REVERSE	41_{16}	One of the SLOW REVERSE trick play states, 41_{16} - 47_{16}
	UNSPECIFIED FAST REVERSE	$4F_{16}$	One of the FAST REVERSE trick play states, 49_{16} - $4F_{16}$
WIND ($C4_{16}$)	As defined by Table 10-7		Any of the values defined for the WIND command may be returned to indicate the corresponding transport state
	STOP EMERGENCY	30_{16}	Unexpected conditions have forced a stop
	STOP CONDENSATION(DEW)	31_{16}	Possible risk of transport damage

If a VCR subunit is unable to precisely determine its transport state, it may return a response that indicates one of the UNSPECIFIED states described above.

A VCR subunit that supports only one speed for any of the trick play groups, slow forward, fast forward, slow reverse, or fast reverse, shall return the corresponding UNSPECIFIED code when in one of the trick play states.

A VCR subunit that supports two or more speeds for any of the trick play groups shall return a code that corresponds to the actual speed of the transport mechanism, as defined by the mapping described in 10 but it may not be equal to the *playback_mode* operand of the PLAY control command that initiated the operation.

> **NOTE:** The value returned in the response is not required to be equal to the value used to activate the transport mechanism. For example, a VCR that mapped 1/30 playback speed to SLOWEST, SLOW FORWARD 6 and SLOW FORWARD 5 might have playback initiated by a PLAY control command that specified SLOW FORWARD 6. A subsequent TRANSPORT STATE inquiry could return SLOW FORWARD 5 to accurately reflect the actual transport speed.

WIND command

The WIND control command is used to control movement of the medium in the transport mechanism. Motion initiated by the WIND control command differs from motion initiated by the PLAY or RECORD control commands in that it may be unmonitored with respect to precise track position. The format of the WIND control command is illustrated by Figure 10-39 below.

	msb							lsb
opcode	WIND ($C4_{16}$)							
operand[0]	subfunction							

Figure 10-39 — WIND control command format

All of the WIND subfunctions typically cause the transport mechanism's playback and record head(s) to be unloaded from the medium and then cause the motion indicated. The value of the *subfunction* operand determines the operation performed by the target, as defined by the table below.

Table 10-7 — WIND subfunctions

Subfunction	Value	Support level	Action
HIGH SPEED REWIND	45_{16}	O	Move the medium toward the beginning of medium as quickly as possible
STOP	60_{16}	M	Halt all transport mechanism motion
REWIND	65_{16}	M	Move the medium toward the beginning of medium
FAST FORWARD	75_{16}	M	Move the medium away from the beginning of medium

If no cassette is loaded, the VCR subunit shall refuse the WIND control command and return a *response* of REJECTED.

WRITE MIC command

The WRITE MIC control command is used to store variable-length data in cassette nonvolatile memory (MIC). A controller is expected to have issued an OPEN MIC control command to the target subunit before performing any MIC read or write operations. The format of the WRITE MIC control command is illustrated by Figure 10-40 below.

	msb							lsb
opcode	WRITE MIC (62_{16})							
operand[0]	data_length							
operand[1]	(most significant byte) MIC_address							
operand[2]	(least significant byte)							
operand[3]	data							
...								
operand[*n*]								

Figure 10-40 — WRITE MIC control command format

The *data_length* field specifies the number of bytes to be written to the MIC.

The *MIC_address* field specifies the starting offset of the data within the MIC.

If *data_length* specifies more bytes than can be accepted by the VCR subunit in a single operation or if the combination of *data_length* and *MIC_address* reference unimplemented memory, the VCR subunit shall reject the WRITE MIC control command.

If an ACCEPTED response frame is returned by the target after a WRITE MIC control command, the response frame confirms the *data_length* and the *MIC_address* but does not echo any of the data bytes written to nonvolatile memory.

In addition to the use of WRITE MIC when *ctype* has a value of CONTROL , WRITE MIC may also be used to query the VCR subunit's capabilities to accept MIC data. The format shown by Figure 10-41 below, when *ctype* has a value of STATUS, is used to determine the maximum length of data that a VCR subunit can write to MIC in a single operation.

	msb							lsb
opcode	WRITE MIC (62_{16})							
operand[0]	FF_{16}							
...								
operand[2]								

Figure 10-41 — WRITE MIC status command format

In the response frame returned by the VCR subunit, *operand[0]* is replaced with the maximum data length that may be written to MIC in a single WRITE MIC operation.

Annex A.AV/C commands in numerical order (normative)

The table below lists all the AV/C commands, in numerical order by *opcode*. For commands that pertain to subunits, *e.g.*, VCR's, columns are headed by a single letter that encodes the type of subunit. The legend for the subunit types follows the table.

Value	Opcode	Unit command	Subunit commands V	Support level (by *ctype*) C	S	N
00_{16}	VENDOR-DEPENDENT	X	X	V	V	V
01_{16}	RESERVE	X	X	O	O	R
02_{16}	PLUG INFO	X	X	–	O	–
10_{16}	DIGITAL OUTPUT	X		O	O	–
11_{16}	DIGITAL INPUT	X		O	O	–
12_{16}	CHANNEL USAGE	X		–	R	R
18_{16}	OUTPUT PLUG SIGNAL FORMAT	X		O	R	O
19_{16}	INPUT PLUG SIGNAL FORMAT	X		O	R	O
20_{16}	CONNECT AV	X		O	O	O
21_{16}	DISCONNECT AV	X		O	–	–
22_{16}	CONNECTIONS	X		–	O	–
24_{16}	CONNECT	X		O	O	R
25_{16}	DISCONNECT	X		O	–	–
30_{16}	UNIT INFO	X		–	M	–
31_{16}	SUBUNIT INFO	X		–	M	–
40_{16}	EDIT MODE		X	O	O	–
45_{16}	PRESET		X	O	O	–
50_{16}	SEARCH MODE		X	–	R	O
51_{16}	TIME CODE		X	R	M	–
52_{16}	ATN		X	R	M	–
53_{16}	RECORDING DATE		X	–	O	–
54_{16}	RECORDING TIME		X	–	O	–
55_{16}	FORWARD		X	R	–	–
56_{16}	BACKWARD		X	R	–	–
60_{16}	OPEN MIC		X	*	R	–
61_{16}	READ MIC		X	R	–	–
62_{16}	WRITE MIC		X	O	O	–
70_{16}	ANALOG AUDIO OUTPUT MODE		X	O	O	–
71_{16}	AUDIO MODE		X	O	O	–
78_{16}	OUTPUT SIGNAL MODE		X	O	M	–
79_{16}	INPUT SIGNAL MODE		X	O	M	–
$B2_{16}$	POWER	X	X	O	O	R
$C1_{16}$	LOAD MEDIUM		X	O	–	–
$C2_{16}$	RECORD		X	*	–	–
$C3_{16}$	PLAY		X	*	–	–
$C4_{16}$	WIND		X	*	–	–
$D0_{16}$	TRANSPORT STATE		X	–	M	O
DA_{16}	MEDIUM INFO		X	–	R	–
DB_{16}	RECORDING SPEED		X	O	O	–

In the preceding table, an asterisk in the support level column indicates that the command operands determine whether the command is mandatory (M), recommended (R) or optional (O).

The legend which defines the subunit type codes is as follows:

Subunit Type Code	Definition
V	Video cassette recorder (VCR)

Annex B. VCR subunit responses (informative)

This annex describes expected behaviors for VCR subunit commands which affect or query the transport state of the VCR, the LOAD MEDIUM, PLAY, RECORD and WIND control commands and the TRANSPORT STATE status and notify commands. The goal is to promote interoperability of different vendors' products.

In addition to the behaviors defined for each of these control commands, it is expected that a VCR subunit return a REJECTED response under the following circumstances:

the VCR subunit or transport mechanism is in an emergency condition,

the loaded medium is positioned at end of medium and a control command that specifies forward motion is received,

the VCR subunit is not powered (although the AV unit which contains the VCR subunit is powered),

the VCR subunit (or the AV unit which contains it) is in a timer recording mode,

the VCR subunit is configured for a different device mode that prevents the operation of the control command, e.g., a camcorder which is in camera, rather than VCR, mode, or

vendor dependent cases that inhibit the control command, e.g., the transition to or from a RECORD mode from any transport state except stopped.

This is not an exhaustive list of conditions that cause the rejection of a LOAD MEDIUM, PLAY, RECORD or WIND control command.

Although the TRANSPORT STATE status and notify commands cannot affect the state of the transport mechanism within the VCR subunit, it is also expected that a REJECTED response be returned after a TRANSPORT STATE command if the VCR subunit is not powered (although the AV unit which contains the VCR subunit is powered).

Annex C. Unresolved issues (informative)

This annex describes areas of the AV/C Digital Interface Command Set that are not yet fully resolved or subject to ambiguous interpretations by implementors.

It is recommended that this informative annex remain a part of the specification until the 1394 Trade Association is able to resolve the ambiguities.

C.1 Command execution model

There is no well articulated model for how commands are to be executed, to what degree command queuing is possible, whether or not response frames are required to be returned in the same order as the corresponding commands were initially issued, how (precisely) response frames are to be correlated with their command frames and so on.

The document editors have been unable to describe consistent behavior for AV devices in this area. The document editors are concerned that these ambiguities will permit varying implementations of both peripherals and host software such that interoperability is compromised.

The matter of response frame identification is troublesome because there is no way to uniquely guarantee that a particular response frame can be matched with its command frame.

C.2 Remote bus resets

In section 6, the behavior of an AV device when it detects a IEEE 1394.0 reset on the local bus is described. The AV device shall discard any in progress transactions without the return of a response frame. This is intended to prevent the return of a response frame to the incorrect controller, since the 6-bit physical ID of the controller may have changed as the result of a bus reset.

The same considerations apply if the controller is located on a remote bus that experiences a bus reset: the physical ID's on that bus may change. This in turn implies that an AV device, if it is to be able to be controlled by a remote controller must have some way to detect the occurrence of a bus reset on the remote bus, in order to be able to discard an in progress transactions.

C.3 Extended subunit addressing - Type and ID

There has been some discussion on the topic of extending the model for subunit addressing, to allow for the definition of greater than 32 subunit types and greater than 5 subunits of a given type within a unit (refer to tables 5 - 1 and 5 - 2). The current version of the AV/C command set has been crafted to allow for this growth, but future work needs to ensure this backward compatibility.

C.4 Notification support

It is desirable for a controller to have the opportunity to receive notification for any change of state in a target. This implies that all commands which cause state changes need to have notification specifications. There are some commands currently defined without notification support as an option, such as DIGITAL INPUT, DIGITAL OUTPUT, and a few others. Future work should focus on defining them.