Appendix

(This Appendix is not part of the American National Standard, but is included for information only.)

A1. A force of 8 to 14 oz (2.2 to 3.9 N) must be exerted on the pressure pad for proper seating against the camera aperture plate.

A2. The two cut-out areas in the pressure pad permit the use of fingers for side-guiding. A force of 1.5 to 2.5 oz (0.42 to 0.70 N) per finger is adequate to ensure picture steadiness if proper take-up torque is applied to the cartridge.

A3. Although sufficient recess from the front surface of the pressure pad to allow for camera claw and camera aperture guide finger penetration, as defined by Dimension C and 3.5, must be provided, additional portions of the pad surface may be recessed also.

A4. The cartridge pressure pad recess, defined by Dimensions D, E, and J, is available for camera claw film transport engagement. The perforation used for the film vertical registration at its stopping position is specified in ANSI PH22.156M-1982, as minus 2 from the perforation adjacent to the image formed by the camera aperture. The horizontal centerline of the camera aperture should nominally coincide with Datum Plane A.

A5. To provide a consistent method of measurement, it is recommended that a cartridge gauging fixture be used which incorporates datum surfaces, a locating pin, and means for exerting locating forces on appropriate surfaces of the cartridge. For pressure pad measurements, a second fixture, incorporating three 0.060-in (1.52-mm) diameter bosses and a means for exerting the appropriate pressure pad seating force, is recommended.

SMPTE RECOMMENDED PRACTICE — RP 138–1986

Control Message Architecture

1. *General*

1.1 Scope. This practice defines the architecture of the control message language used within a general-purpose communications channel of an interface system which transports data and control signals between equipment utilized in the production, post-production, and/or transmission of visual and aural information.

It is intended that the language described in this practice be utilized when constructing messages used as part of an overall system, allowing interconnection of programmable and nonprogrammable equipment as required to configure an operational system with a defined function, and to allow rapid reconfiguration of a system to provide more than one defined function utilizing a given group of equipment.

1.1.1 Control message language is composed of vocabulary, syntax, and semantics expressed in terms of tokens, rules, and actions, respectively.

1.1.2 The primary intent of this practice is to define the architecture of the messages to be transmitted within the supervisory protocol of the communications channel for the purpose of controlling equipment by external means. Syntax is the set of rules which shall be applied to the vocabulary (tokens) to construct control messages. (The content of the vocabulary and its semantics, being specific to the type of generic equipment, is defined elsewhere.) This practice, or sections thereof, may be applied to the interconnection of elements within an item of equipment.

1.2 Definitions. For the purpose of this practice, the following definitions shall apply:

Virtual Machine: A logical device consisting of a single device or a combination of devices that respond in essence or effect as a generic type of equipment; e.g., VTR, video switcher, telecine, etc.

Virtual Circuit: A transparent, logical, communications connection between virtual machines. The communications path, in reality, passes through other levels and is propagated over a physical medium.

2. *Message Structure*

2.1 Architecture. The message architecture described in this practice is prepared broadly on the principals of communications levels. This architecture follows a logical structure and is defined in terms of a virtual machine. Messages are of variable length according to function. Complex functions may be divided into basic functions, transmitted as a sequence of shorter messages for execution in the virtual machine.

2.2 Virtual Machine. All messages pertaining to generic types of equipment shall be defined in terms of the virtual machine. Utilization of the virtual machine concept in defining messages provides a message architecture that is independent of machine-specific characteristics.

3. *Control Message Classification*

3.1 Control messages are classified in accordance with Fig. 1.

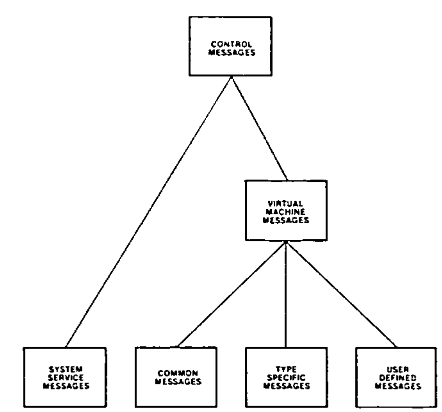


Fig. 1
Message Classification

SOCIETY OF MOTION PICTURE AND TELEVISION ENGINEERS
595 West Hartsdale Avenue, White Plains, NY 10607, (914) 761-1100

Approved 1 May 1986

3.1.1 Virtual machine messages are used to pass commands and responses between virtual machines. Virtual machine messages are those initiated by a controlling device with responses originating in the controlled device. Receipt of a virtual machine message shall result in a defined action and/or response by the virtual machine.

Virtual machine messages may be subdivided into:

3.1.1.1 Common messages whose coding is reserved to provide for functions of general application; e.g., procedures, reference time functions, and reset.

3.1.1.2 Type-specific messages are applicable to specific generic categories of equipment.

3.1.1.3 User-defined messages implement special functions which are not included in the type-specific message set.

3.1.2 System-service messages are messages other than virtual machine messages.

3.2 Virtual Machine Message Subsets. A separate and distinct subset of virtual machine messages shall be specified for each type of virtual machine (VTR, telecine, audio tape recorder, graphics *generator, etc.)*. *Said subset, termed a dialect,* shall comprise common messages, type-specific messages and, optionally, user-defined messages.

3.2.1 Common Messages: Resident machine messages which are in all virtual machine dialects but not necessarily operative in all virtual machines, whose coding is reserved to provide for functions of general applications.

3.2.2 Type-Specific Messages: Virtual machine messages *which are defined in virtual machine dia*lect recommended practices.

3.2.3 User-Defined Messages: Virtual machine messages which are unique to the type (manufacturer, model, version, S/N, etc.) of the specific machine. Although the definition and/or documentation of user-defined messages is considered outside the scope of this practice, the structure of such messages shall conform to the message architecture as defined herein.

4. *Control Message Construction*

4.1 Syntax. System service and virtual machine messages are uniformly constructed with the following syntax:

MESSAGE = KEYWORD (+ ARGUMENT)

where the keyword characterizes the function to be performed and the argument contains the parameters, where necessary, to perform that function.

A parameter has the following syntax:

PARAMETER = (NAME +) VALUE(S)

The name may be implied with the use of specific keywords and, in such cases, is therefore not required. The length and format of the value (or values) is defined by the name *(or implied name)*. No restriction is placed on the possible concatenation of parameter values.

4.2 Message Formats. All control messages are formed as groups of integral bytes. The first byte of each message shall be the keyword. A keyword specification defines the format of its arguments; although no mathematical relationship is intended between the bit pattern of the keyword and the format. Messages are constructed in one of the following formats:

```
Format 1  Message    = <Keyword>
Format 2  Message    = <Keyword> <Parameter List>

        where:
   <Parameter List>  = <Parameter>
        or:
   <Parameter List>  = <Begin> <Parameter Group> <End>

        where:
   <Parameter Group> = <Parameter>
        or:
   <Parameter Group> = <Parameter Group> <Parameter>

        where:
   <Parameter>       = <Parameter Value> . . . <Parameter Value>
        or:
   <Parameter>       = <Parameter Name>
                     = <Parameter Value> . . . <Parameter Value>
```

The appropriate message format can be selected by means of the decision tree given in Fig. 2.

5. *Message Coding*

5.1 Identical or similar functions on equipment of differing generic type should be effected by the same keyword bit pattern.

5.2 Parameter Values. Messages may contain parameters as an essential part. All parameters are classified as follows:

5.2.1 Logical Parameter Values. Parameters representing any abstract function(s) that may be expressed by a simple binary state of 1 (true) or 0 (false) such as tally on/off or yes/no.

The minimum code length for a single logical parameter is one byte. Individual logical parameters can be assembled, where applicable, into groups to form bit-specific bytes for transmission purposes.

5.2.2 Numerical Parameter Values. Parameters representing a numeric value and consisting of the following:

Unsigned number parameters:
Parameters representing any numeric value without polarity.

Signed number parameters:
Parameters representing any numeric value with polarity.

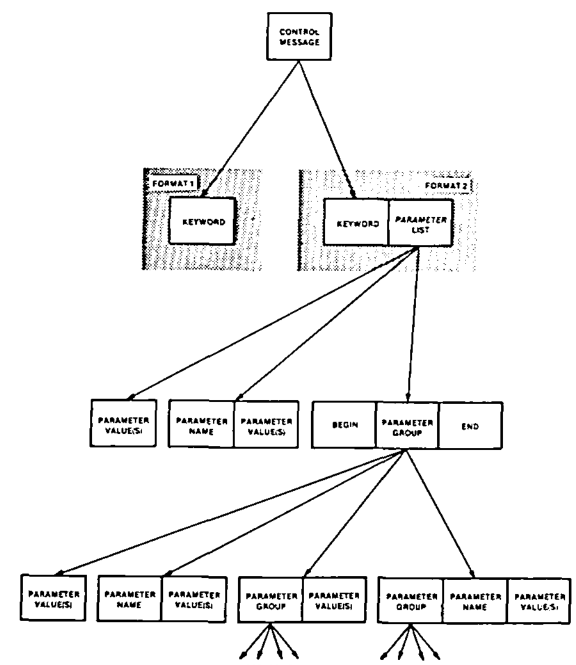


Fig. 2
Decision Tree

Time-Code Parameter Values: Time is indicated as a 4-byte quantity. Parameters representing hours, minutes, seconds, and frames are expressed in BCD in that order. The Hex "40"-bit of the frame's byte will be set to one (1) in Drop-Frame Compensated Mode. In Nondrop Frame Uncompensated Mode and all other time code standards, this bit will be zero (0). In all standards, the Hex "80"-bit of the second's byte will be set to zero (0) to indicate monochrome field 1 or color fields 1, 3, 5, or 7. This bit set to one (1) will indicate monochrome field 2 or color fields 2, 4, 6, or 8. Unused bits are reserved and are set to zero (0) until defined. (See Fig. 3.)

High-Resolution Time Code Parameter Values: High-resolution time is indicated as a 6-byte quantity. The first 4 bytes are exactly the same as time parameter values. The two remaining bytes express fractions of frames as a 16-bit binary unsigned number. (See Fig. 3.)

5.2.3 Literal Parameters are parameters based, in general, on ASCII characters.

5.2.4 Raw Data Parameters are parameters based on a free-form data stream. Raw data parameters must provide for byte transparency to the lower layers. The first byte of a raw data parameter shall be a byte count.

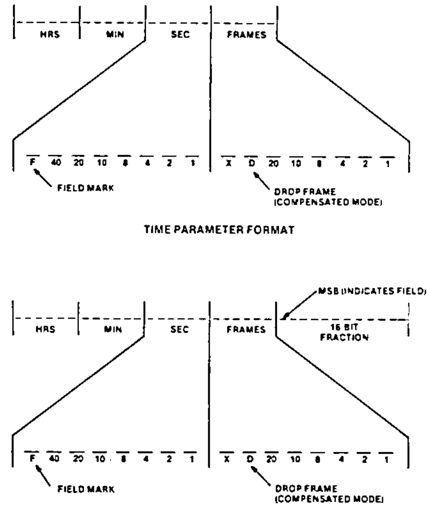


Fig. 3
High-Resolution Time Parameter Format

SMPTE RECOMMENDED PRACTICE

RP 139-1986

Tributary Interconnection

1. *General*

1.1 Scope. This practice describes the mechanism for the transfer of control messages between tributaries used within a general-purpose communications channel of an interface system which transports data and digital control signals between equipment utilized in the production, post-production, and/or transmission of visual and aural information.

It is intended that the mechanism described in this practice be utilized when transferring control messages between tributaries used as a part of an overall system. The tributaries may be located either within a local network or on separate local networks which are interconnected by means of gateways and an interconnection bus.

It is further intended that this mechanism, when used as part of an overall system, shall allow the interconnection of programmable and non-programmable equipment as required to configure an operational system with defined function, and will allow rapid re-configuration of a system to provide more than one defined function utilizing a given group of equipment.

1.1.1 The message transfer mechanism makes use of virtual circuits, linkage directories, and system service messages (defined below).

1.1.2 The primary intent of this practice is to define the mechanism enabling the transfer of messages between tributaries for the purpose of controlling equipment by external means.

1.2 Definitions. For the purposes of this practice the following definitions shall apply:

Virtual Machine: a logical device consisting of a single device or a combination of devices that respond in essence or effect as a generic type of equipment, e.g., VTR, video switcher, telecine, etc.

Virtual Circuit: A transparent, unidirectional, logical communications connection between virtual machines. The communications path, in reality, passes through other levels and is propagated over a physical medium.

2. *Interconnection within a Local Network*

2.1 Message Transfer. The mechanism for message transfer between tributaries is based broadly on the principles of communications layering and makes use of virtual circuits. This allows for the establishing of, and breaking down of, multiple links between the tributaries. System service messages perform this function.

A 'linkage directory' is established within the bus controller for each working session. The directory is considered to be a system service feature and provides for the establishment of multiple virtual circuits through the network.

2.2 Linkage Directory. The linkage directory shall establish a relationship between virtual machines, i.e., a virtual circuit. Establishment of the linkage directory shall be completed as the initial task in each working session. The linkage directory resident within the system service level of the bus controller binds message 'sources' and 'destinations.'

Linkage information may originate in any application level, and shall effect directory construction within the system service level of the bus controller. Linkage messages are reserved messages within the system service sub-set of all message dialects; they establish and disconnect virtual circuits within the network.

The bus controller, on receipt of a transmission request from the supervisory level of any tributary, will identify the destination tributary by reference to the linkage directory; acting as an intermediary it will forward the message as directed.

2.3 Multiplexing within Tributaries. Tributaries, in general, have a single supervisory level address, and a single physical connection end point to the bus. Alternative multiplexing mechanisms, as described below, enable multiple virtual circuits to pass through any single connection end point.

2.3.1 Multiple, logically independent virtual machines, each with a unique supervisory level address, may be attached to the communications channel through a common connection end

Approved 1 May 1986

point. Multiplexing is then performed by multiple polling of the addressing entity residing within the supervisory level (Fig. 1).

(It may be noted that any individual tributary address may achieve a higher priority — and hence an improved response time at the expense of that of the remaining tributaries — by being allocated more than one poll within each poll sequence.)

2.3.2 Alternatively, a single supervisory level address may be multiplexed to multiple logically independent virtual machines, with selection being performed by a logical switch residing within the entity of the destination tributary system service level. (See Fig. 2.)

The required virtual machine is selected from those associated with the single supervisory level address, by means of a system service 'virtual-machine-select' message. (See 3.4 below.) This is transmitted from the bus controller under the direction of the linkage directory held within its system service level, to the destination tributary system service level, immediately prior to the transmission of any control message, or sequence of control messages, destined for that specific virtual machine.

The selected routing will remain in existence until receipt, by the system service level, of a new virtual-machine-select message thereby minimizing the message traffic on the communications channel.

2.3.2.1 The reverse route of each virtual circuit, when required, will be selected similarly by the logical switch resident within the entity of the system service level of the multiplexed tributary. This selection is performed on receipt of a control or response message from any one of the virtual machines attached to the system service level of the tributary. The system service level will then instruct its supervisory level to transmit the appropriate 'virtual-machine-select' message to the supervisory, and hence the system service, level of the bus controller.

2.3.2.2 System service level group 'Assign' and 'De-assign' commands shall be used to assemble/disassemble groups of virtual machines within the system service level, from those associated with a single supervisory level address, for simultaneous control purposes.

Virtual circuits employing virtual group identifiers shall be recorded as additional entries within the bus controller linkage table.

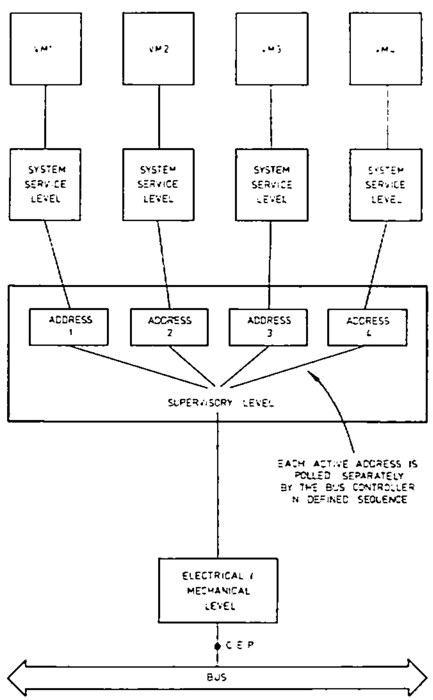


Fig. 1
Multiplexing within Supervisory Level

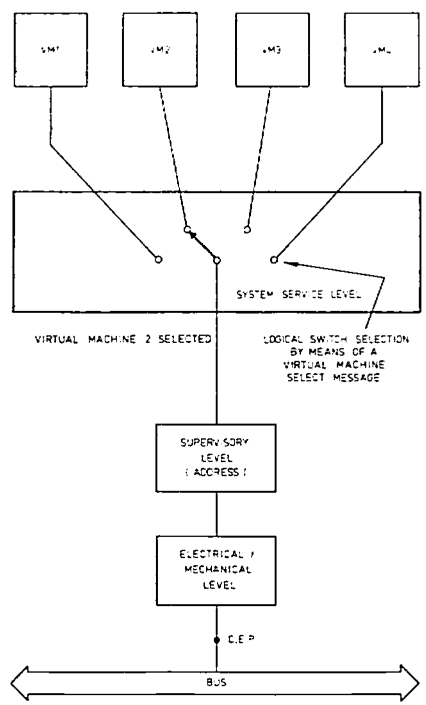


Fig. 2
Multiplexing within System Service Level

2.3.3 It should be noted that a bus overhead exists in each method of virtual circuit multiplexing. Where the multiplex is to take place within the supervisory level (2.3.1), the overhead will take the form of additional polls in each cycle.

System service level multiplexing (2.3.2) introduces an additional control message (the virtual-machine-select message) prior to each virtual machine message, or series of virtual machine messages, destined for an alternative virtual machine.

The choice of multiplexing mechanism, where used, rests with the system designer in recognition of specific design considerations.

2.4 Forbidden Configurations. Some virtual circuit configurations may be forbidden due to the function of the particular tributary, i.e., the functions of the tributaries are incompatible. Checking mechanisms should be employed to ensure that illegal virtual circuits cannot be established. Most of the checking would be performed in the system service level according to predefined rules within the particular network. Some rules could be readily derived from the type of tributary (built in) while others may be imposed by the user or system designer.

3. *System Service Messages*

System service messages are messages contained in the system service sub-set of all message dialects and shall be used to command the performance of system functions. These functions include, but are not limited to:

3.1 Segmentation and Re-assembly. These processes enable the transfer of messages which exceed the maximum supervisory level message block length (see Fig. 3a). The parsing mechanism for segmentation and blocking is described by the state diagram given in Fig. 4.

3.1.1 A data segment shall take the following form (see Fig. 3b):

1st Byte:	Keyword SEGMENT
2nd Byte:	Number of segments remaining: last segment is 0; segment count shall be sent in sequentially descending order.
Remaining Bytes:	Segment data. No further message shall follow a data segment message within a single supervisory level block.

3.2 Blocking and De-blocking. These processes enable the concatenation of messages within a single supervisory level message block.

3.2.1 A Data Block shall take the following form (see Fig. 3c):

1st Byte:	Keyword BLOCK.
2nd Byte:	Byte count (N), where N is the number of bytes in the block data.
Remaining Bytes:	Block data.

3.2.2 The supervisory level shall transfer the byte count to the system service level.

3.3 Establishment of Virtual Circuits. This process is effected through the management of the linkage directory contained within the bus controller.

3.4 Selection of a Virtual Machine. This process enables the selection of a virtual machine from those previously assigned to a tributary.

3.5 Tributary reset. This command returns the tributary to its power-up default state.

3.6 Group Assign/De-assign. These commands establish/break down system service level groups of virtual machines for joint control purposes.

3.7 Virtual Group Assign/Deassign. These commands establish/break down supervisory level groups of tributaries for joint control purposes.

4. *Interconnection of Local Networks*

4.1 Interconnection Bus. Interconnection of individual local networks shall be by means of an interconnection bus (see Fig. 5). Linking of the local network to the interconnection bus shall be by means of a GATEWAY.

ISO 3309 and 4335 (HDLC), in accordance with CCITT Recommendation X.25 — LAPB, shall be used for the data link layer protocol between the gateway and the interconnection bus coupler; the physical link layer shall be as specified in CCITT Recommendation X.21.

4.2 Gateway. The gateway is a logical device whose task is to transfer messages between a local network and an external interconnection bus coupler. The gateway provides for the interchange of messages between multiple local networks.

The gateway will maintain a linkage directory in its system service level. The linkage table will allow the gateway to be seen by the bus controller as a set of 'virtual' tributaries linked by virtual circuits.

The gateway will provide for all protocol conversions required to convert from the interface bus supervisory and electrical/mechanical level standards as specified in SMPTE Recommended Practice on Supervisory Protocol for Digital Control Interface, RP 113-1983, and American National Standard for Television — Digital Control Interface — Electrical and Mechanical Characteristics, ANSI/SMPTE 207M-1984, respectively, to the HDLC data link and X.21 physical link layers.

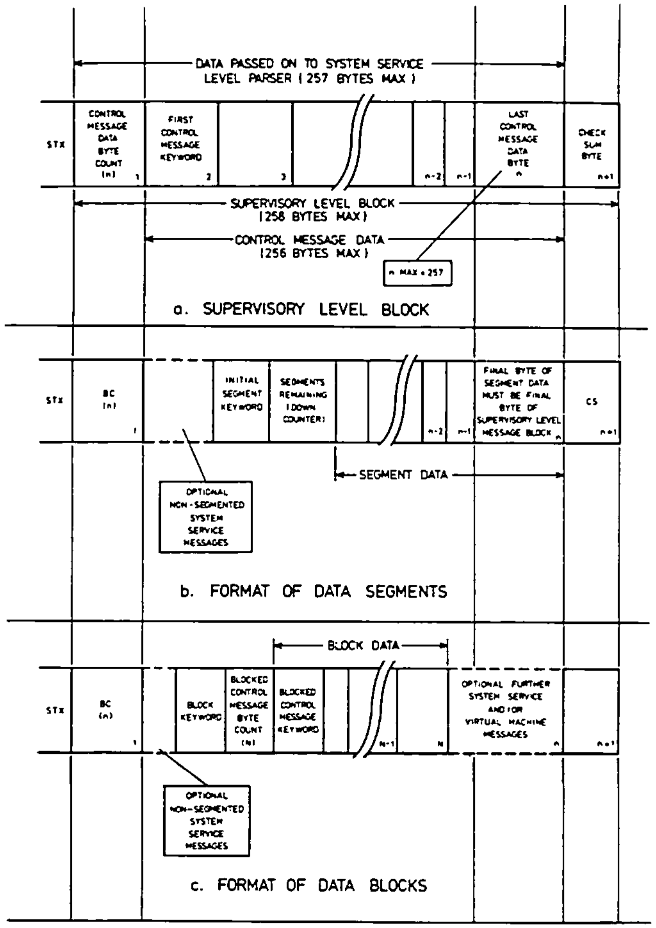


Fig. 3
Data Segments and Blocks

The gateway will provide decoding of group addresses provided for in the supervisory level (SMPTE RP 113-1983) and will forward messages addressed to these groups over the interconnection bus as discrete individual select addresses. Where more than one 'external' tributary is addressed by a group message, the individual messages to all such tributaries shall be dispatched sequentially as individual messages from the gateway. Translation takes place in the system service level of the gateway. The functional structure of the gateway is shown in Fig. 6.

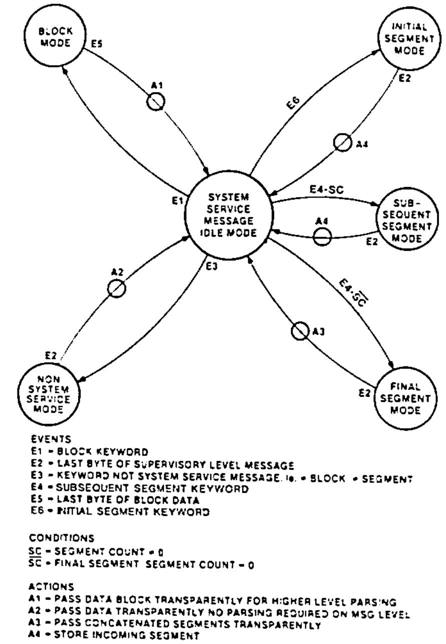


Fig. 4
Segmentation/Blocking State Diagram

5. *Guidelines*

This section gives a typical example of virtual machine selection when using the multiplexing technique detailed in 2.3.2. It encompasses operations in both the system service and supervisory levels and thus includes features described in SMPTE RP 113-1983.

In 5.1, the procedure is described in broad outline; in 5.2, the same example is dealt with in more rigorous detail.

5.1 In this broad outline, the form of the messages is not defined precisely but is given only as an illustration of the function to be performed.

(A) Assume that three control panels are linked to the local network through a single tributary address and connection end point as shown in Fig. 7.

During the assignment process, the control panels CP1, CP2, and CP3 have been associated with VTR, telecine and still store, respectively, via virtual circuits (1), (2) and (3).

(B) Assume further that a VTR command has just been issued by CP1 and a telecine PLAY command is now required.

(C) The following linkage message must be issued by the system service level of the control panel tributary:

[Virtual-machine-select] [2]

This changes the virtual machine selection from virtual machine [1], (VTR), to virtual machine [2], (telecine).

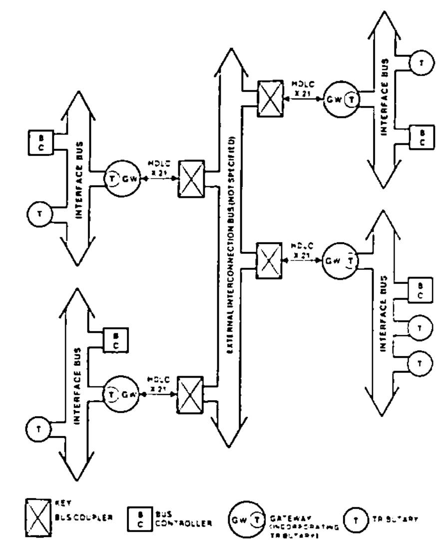
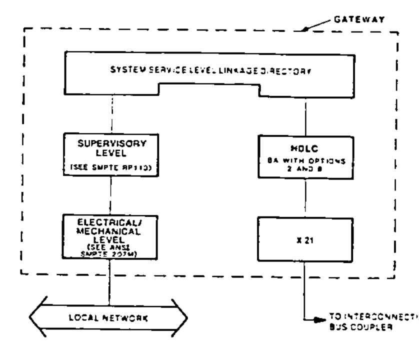


Fig. 5
Local Network Interconnection



NOTE:
THE HDLC/X.21 DEFINITION SHALL BE IN ACCORDANCE WITH THE CCITT X.25-LAPB

Fig. 6
Gateway Functional Structure

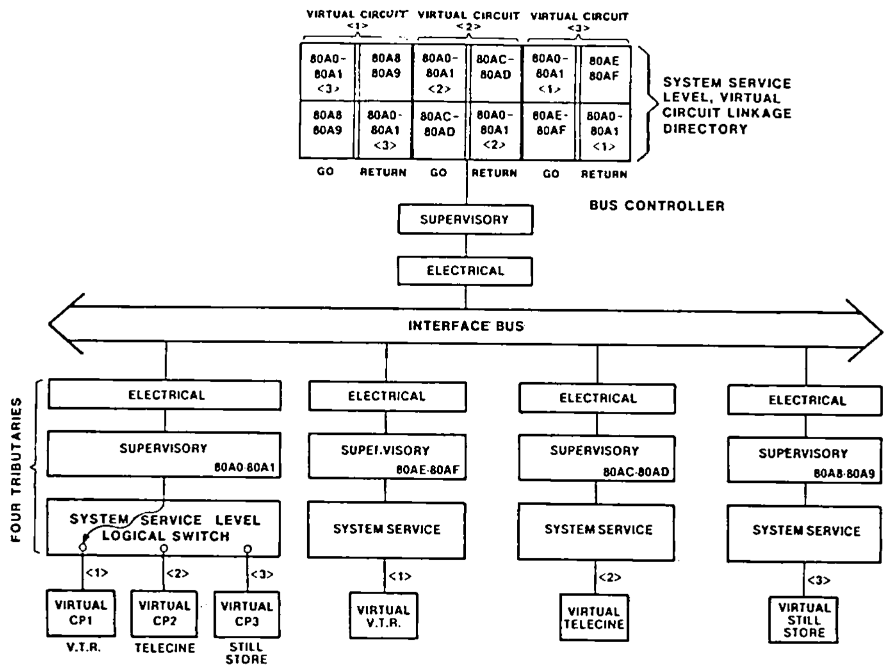


Fig. 7
Example of Virtual Circuit Select Mechanism

(D) The control panel virtual machine then issues the control message:

[PLAY]

This causes the telecine virtual machine to change to the play state.

Any subsequent messages from the control panel to the telecine will be transferred without any further linkage messages, e.g., the control message [STOP].

A [NEXT SLIDE] command for the still store virtual machine would, however, require:

1. [Virtual-machine-select] [3] and
2. [NEXT SLIDE]

in order to re-select the virtual machine CP3.

5.2 In this more rigorous treatment of the example given in 5.1, it is assumed that the three control panel virtual machines, CP1, CP2, and CP3, are linked to the interface bus through the single tributary address [80A0/80A1] and connection end point.

5.2.1 A [START] command from the telecine control panel virtual machine CP2 attached to tributary 80A0/80A1 is to be sent by virtual circuit [2] to the telecine virtual machine connected to tributary 80AC/80AD. A possible message exchange might be:

(A) Telecine control panel virtual machine, (CP2), passes [START] command to system service level of tributary 80A0/80A1.

(B) 80A0/80A1 system service level instructs supervisory level to raise the service request flag (SVC).

(C) The bus controller, as part of its normal poll sequence, polls 80A1; and receives [SVC].

(D) The bus controller issues select address 80A0; it then sends [TEN] to 80A0/80A1 supervisory level.

(E) 80A0/80A1 supervisory level sends:

[STX] [byte count (BC)] [virtual-machine-select] [2] [block check (B.CK)]

to the bus controller (see Note 1).

(F) The bus controller responds with [ACK] and a further [TEN]. (Since the last message was a 'virtual-machine-select' message, a further virtual machine control message is expected by the bus controller (see Note 1).

(G) The supervisory level of the tributary 80A0/80A1 sends:

[STX] [BC] [START] [B.CK]

to the bus controller (see Note 1).

(H) The bus controller system service level identifies the destination of [80A0/80A1 — virtual-machine 2] from its linkage directory. The address is found to be 80AC/80AD.

(I) The bus controller issues [BREAK] followed by the select address 80AC.

(J) 80AC/80AD tributary supervisory level responds with [ACK].

(K) The bus controller then sends:

[STX] [BC] [START] [B.CK]

to tributary 80AC/80AD.

(L) The supervisory level of tributary 80AC/80AD responds with [ACK] and passes the control message to the system service level parser.

(M) The system service level parser passes the [START] command to the telecine virtual machine.

Note 1: The messages in (E) and (G) might be concatenated into the single 'hybrid' command:

[STX] [BC] [virtual-machine-select] [2] [START] [B.CK]

in order to limit protocol overhead. In this case the message contained in (F) would not be necessary.

5.2.2 A tally response [STARTED] from the telecine virtual machine tributary 80AC/80AD is to be sent to telecine control panel virtual machine CP2 attached to the interface bus through tributary 80A0/80A1.

(A) The telecine virtual machine passes the [STARTED] tally to the system service level of tributary 80AC/80AD.

(B) The system service level instructs the supervisory level of 80AC/80AD to raise the service request flag (SVC).

(C) The bus controller, as part of its normal poll sequence, polls 80AD and receives [SVC].

(D) The bus controller issues the select address 80AC, followed by [TEN] to the supervisory level of 80AC/80AD.

(E) The bus controller receives the tally:

[STX] [BC] [STARTED] [B.CK]

from 80AC/80AD.

(F) The bus controller system service level determines the destination (80A0/80A1 — virtual machine 2) from its system service level linkage directory.

(G) The bus controller issues [BREAK] and the select address 80A0.

(H) 80A0/80A1 supervisory level responds with [ACK].

(I) The bus controller sends:

[STX] [BC] [virtual-machine-select] [2] [B.CK]

to tributary 80A0/80A1 (see Note 2).

(J) The tributary 80A0/80A1 responds with [ACK], and sets the logical switch in its system service level to select telecine control panel virtual machine CP2.

(K) The bus controller sends tally:

[STX] [BC] [STARTED] [B.CK]

to tributary 80A0/80A1 supervisory level (see Note 2).

(L) The supervisory level of tributary 80A0/80A1 responds with [ACK] and passes the control message to the system service level parser.

(M) The system service level parser passes [STARTED] tally to the telecine control panel virtual machine CP2.

Note 2: The messages in (I) and (K) might be concatenated into a single hybrid command thus:

[STX] [BC] [virtual-machine-select] [2] [STARTED] [B.CK]

in order to limit protocol overhead.

5.2.3 It should be noted that further commands to the same virtual machine, and which follow immediately on the sequences detailed in 5.2.1 will omit steps (E) and (F) since no further changes are needed in the virtual machine selection.

Similarly, 5.2.2 steps (I) and (J) will be omitted under the same circumstances.