HDMI-Generator Option Overview

HDMI is a derivative of DVI, which adds many new features to the popular TMDS-based uncompressed high-speed digital video interface.

On the video side, HDMI defines extra pixel encodings - beyond the traditional RGB 8-bits per component 4:4:4-sampled variety. YCbCr color difference encoding is now possible with two sampling options: 4:4:4 and 4:2:2. Support for 8-bits per component continues with YCbCr 4:4:4 encoding, while expanded color depth options of 8, 10, or 12-bits per component are made possible by the alternate YCbCr 4:2:2 color subsampling mode.

HDMI also adds a new data path to the TMDS interface. The transport of digital data packets from generator (source) to display (sink) is now possible. This paves the way for three additional features: digital audio, AV mute, and EIA/CEA-861-C InfoFrame based display-mode automation.

Major Blocks

Figure 1 shows the major blocks of the new HDMI generator option. Alongside each block are important parameters related to that block.

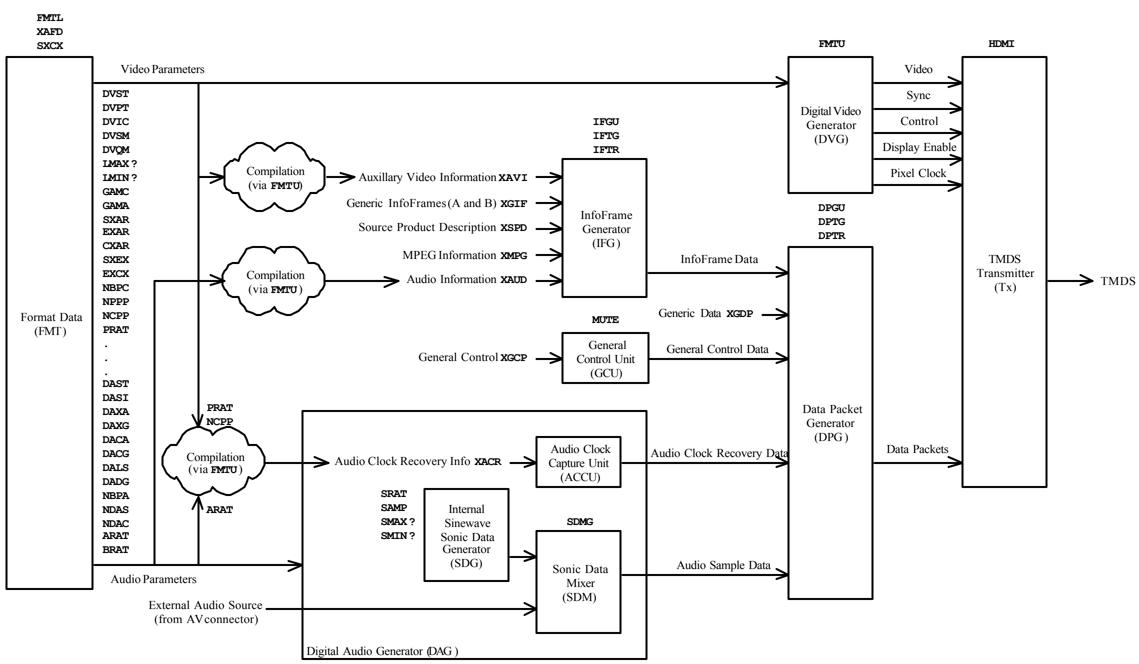


Figure 1 - HDMI Option Block Diagram

On the left and at the top-right, of the diagram, are the familiar format data (FMT) and digital video generator (DVG) blocks. These two blocks are basic to any video signal generator.

HDMI extends the basic video generator, with added blocks that produce various types of data - in addition to video. At the bottom of **Figure 1** is a new digital audio generator (DAG) block, which generates audio samples and audio clock recovery data. Just above the DAG are the InfoFrame generator (IFG) and general control unit (GCU) blocks, which generate InfoFrames and general control data, respectively. A generic data packet buffer (XGDP) is available for sourcing highly customized arbitrary data.

All of these data sources feed a data packet generator (DPG), which combines and packetizes information for transmission by the transmitter (Tx).

To the immediate right of the format data block, in **Figure 1**, is a partial list of format parameters. These parameters are specific to the new HDMI feature set. At the top are some video parameters. At the bottom are many new audio parameters. The two sets of format parameters are used to control the DVG and DAG, respectively. They are also used as the basis for the compilation of various clusters of data that are eventually transmitted to the display in order to effect audio reference clock regeneration and display-mode automation.

Transmitter (Tx)

At the far right of **Figure 1** is a new HDMI/DVI transmitter (Tx), which can output either HDMI or DVI compatible TMDS video as needed. HDMI video is different than DVI video in that it has an advanced protocol that allows both data and video to be transported over a common TMDS physical interface (PHY). Sending an HDMI signal to a DVI-only display will usually yield a noisy picture or no picture at all.

HDMI/DVI TMDS Protocol Selection

The type of video that is output by the Tx can be controlled from either the serial command line interface or the front panel controls of the generator.

On the command line, output protocol is controlled by the `DVPT` format parameter, which is shown just above the block Tx in **Figure 1**. Setting the `DVPT` parameter to 2 and issuing a `FMTU` (or `ALLU`) command causes the output signal protocol to switch to HDMI, while setting `DVPT` to 1 causes a DVI compatible protocol to be output. All of the formats in our library set `DVPT` to 0. This is a default setting, which lets the output protocol be determined, on a global basis, by the type of hardware that is present. If HDMI-capable generator hardware is detected and a digital video signal type is specified, then the generator will output HDMI. Other methods for controlling output protocol are discussed next.

Force DVI Option

Generators, that have the HDMI option installed, output HDMI video by default. A special power-on option called “DVI (not HDMI)” (a.k.a. “DVI friendly”) is available that can be used to force the Tx to always output DVI-compatible video - instead HDMI. The option is enabled either by holding down the R and G keys at power-on or by setting mask bit-15 of the `SROP` parameter. As with all power-on options, once set, this option remains set – even if the generator power is cycled.

Signal Interface/Protocol Toggle

Once the generator is running, the output TMDS protocol can be established manually using the R and G multi-key rollover. Note that, in older firmware revisions, the R and G multi-key rollover was used for picture memory depth selection. Modern firmware automates the selection of memory depth – freeing-up the R and G key operation for use as an output protocol control. When using the R and G rollover, it is important that the “Protocol, Encoding, and Sampling Status Display Option” is enabled as well. If not, you will not be able to see what protocol you are outputting and will not know when to stop toggling. This option is discussed next.

Protocol, Encoding, and Sampling Status Display Option

When working with multiple TMDS protocols and color encoding methods, it is a good idea to turn-on the "display status" option by holding down the STEP, G, and B keys at power-on. This will cause additional information to be displayed on the front panel LCD of the generator - just to the left of the format name - indicating the current protocol and color encoding methods that are in use. The letter "H" will appear whenever the HDMI protocol is being output, while the letter “D” will appear in the case of DVI. Whenever the component trio is RGB, the letter “C” will appear, while the letter letters “Y” or “y” indicate the presence of YCbCr color difference video encoding. In the case of color difference video, an uppercase “Y” indicates 4:4:4 color sampling, while a lowercase “y” signals 4:2:2 color sampling.

HDMI Command/Query

`HDMI` is a parameter that can be queried in order to find out if the current output protocol is HDMI. A value of 1 signifies HDMI protocol, while the value 0 indicates that some other protocol (e.g. DVI) is being used. The `HDMI` command can be used to automatically set the `DVPT` digital video protocol parameter to an appropriate value. Table 1 shows how the HDMI command affects the `DVPT` parameter as a function of the hardware interface being used.

Table 1 - HDMI Command Logic

Interface Hardware Used	HDMI Argument Given	DVPT Setting
DVI-only	0	0
	1	2
HDMI	0	1
	1	0

Note that the `DVPT` parameter is set to 2, whenever DVI-only interface hardware is being used and the “`HDMI 1`” command is given. In this case, the generator will report an error, because the DVI-only hardware does not support the HDMI protocol. On generators that have both DVI-only and HDMI interfaces, the `DVSI` parameter is used to determine which interface hardware is being used.

EDID-based HDMI/DVI TMDS Protocol Selection Via Automatic Hot-plug Format List Option

Another special power-on option called “hot-plug formats” causes the generator to respond to hot-plugs by reading the EDID of a display (or sink) and creating a custom list of formats (on the FORMAT knob), containing all of the timings that the display (sink) says that it supports. The “hot-plug formats” option can be activated by holding down the R and ACS keys at power-on (or by setting mask bit-14 of the `SROP` parameter). Once the generator is running with the “hot-plug formats” option enabled, you can temporarily return the standard format library to the FORMAT knob by executing an R and ACS multi-key rollover. Note that, when you do this, the hot-plug process no longer manages the current setting of the HDMI TMDS protocol parameter. Therefore, if you switch displays (sinks) with the “hot-plug formats” disabled, the output TMDS protocol may not automatically switch to what the current display requires. In this case, the “signal interface/protocol toggle” can be used to switch the output protocol manually.

Canceling Special Power-On Key Hold Options

The only way to remove special options (when using the front panel controls) is to hold down the ACS, DCS, and DSS keys at power-on. This activates the "no special modes" feature, which removes all of the special modes that may have been previously set. Special options can also be selectively set and (or) reset using the `SROP` serial command. The status of all special options can be displayed by selecting the "GenOps" image. The "GenOps" image also lists all of the key operations just discussed – in case you forget what keys do what.

Digital Video Generator (DVG)

Table 2 summarizes fourteen of the digital video types that can be output by the generator. Other types are possible, but are not listed here.

Table 2 - Digital Video Signals

<table>
<tr><th>Components</th><th>Color Model</th><th>Sampling</th><th>NBPC</th><th>DVST</th><th>DVSM</th><th>DVQM</th><th>GAMC</th><th>GAMA</th></tr>
<tr><td>RGB</td><td>Linear</td><td rowspan="2">4:4:4</td><td rowspan="2">8</td><td rowspan="2">10</td><td rowspan="2">4</td><td>0</td><td>0</td><td>X</td></tr>
<tr><td>RGB</td><td>Gamma Corrected</td><td rowspan="13">2</td><td rowspan="13">1</td><td rowspan="13">2.222</td></tr>
<tr><td rowspan="12">YCbCr</td><td rowspan="4">Legacy "240M" HDTV
SMPTE 260M-1999
Table 1</td><td rowspan="3">4:2:2</td><td>12</td><td rowspan="4">13</td><td rowspan="3">2</td></tr>
<tr><td>10</td></tr>
<tr><td>8</td></tr>
<tr><td>4:4:4</td><td>8</td><td>4</td></tr>
<tr><td rowspan="4">SDTV
ITU-R BT.601-5
Table 3, Item 7</td><td rowspan="3">4:2:2</td><td>12</td><td rowspan="4">14</td><td rowspan="3">2</td></tr>
<tr><td>10</td></tr>
<tr><td>8</td></tr>
<tr><td>4:4:4</td><td>8</td><td>4</td></tr>
<tr><td rowspan="4">Modern" 274M" HDTV
ITU-R BT.709-5
Part 1, Section 6.10</td><td rowspan="3">4:2:2</td><td>12</td><td rowspan="4">15</td><td rowspan="3">2</td></tr>
<tr><td>10</td></tr>
<tr><td>8</td></tr>
<tr><td>4:4:4</td><td>8</td><td>4</td></tr>
</table>

Digital Video Signal Type (DVST)

The `DVST` digital video signal type command/query simultaneously selects both the trio of components, as well as the color model (matrix) that are used. Both RGB and YCbCr component trios are available. The RGB color model is quite simple, because it is the space in which test images are natively rendered. RGB may be gamma corrected. When it is, the "Any Other" equation of Table 4 is used. In the case of YCbCr, any one of three different color models (RGB-to-YCbCr matrices) can be selected. The SMPTE 260M color model is available for use with legacy 1035i (SMPTE 240M) HDTV systems, while the ITU-R BT.601 and ITU-R BT.709 color models are available for use with standard definition (SDTV) and modern high-definition (HDTV) television systems, respectively.

Digital Video Sampling Mode (DVSM) and Number of Bits per Component (NBPC)

The `DVSM` digital video sampling mode command/query comes into play whenever any of the YCbCr color difference signal types are selected. This parameter can be used to control how the color difference components (CbCr) are sampled and routed through the TMDS interface. Setting DVSM to 4 enables 4:4:4 sampling. In this mode, both luminance and color difference components are sampled at the pixel rate. As an alternative, DVSM can be set to 2, thereby enabling 4:2:2 sampling. With 4:2:2 sampling, the color difference components (CbCr) are sampled at half the pixel rate, while the Y component continues to be sampled at the full rate. YCbCr 4:2:2 color subsampling reduces the amount of spatial color data that must pass through the TMDS interface - thereby freeing-up bandwidth, which can then be optionally reallocated to increase the precision of individual samples (from 8 to as many as 12-bits/component). With 4:2:2 sampling, HDMI multiplexes the most significant 8-bits of Cb and Cr on TMDS channel 2, while TMDS channel 1 continues to carry the 8-most significant bits of the Y component. This arrangement leaves TMDS channel 0 open to carry four additional bits of Y and C information during each sample. The eight additional bits mean that each Y and C components have an additional 4-bits each for a total of 12-bits of precision. The number of bits per component NBPC command/query can be used to set color depth, in this case, to 8-bit, 10-bit, or 12-bits per component. When 4:2:2 sampling is used and color depth (NBPC) is set for 8 or 10-bits, unused bits are forced to zero.

Digital Video Quantization Mode (DVQM)

Component values can be quantized in a variety of ways. Table 3 shows how the gamut of each digital component varies as a function of the settings of the DVQM digital video quantization mode and NBPC number of bits per component parameters.

Table 3 - Digital Video Quantization Modes

Digital Video Quantization Mode	Components	NBPC=6 gamuts[1]		NBPC=8 gamuts		NBPC=10 gamuts		NBPC=12 gamuts	
		LMIN	LMAX	LMIN	LMAX	LMIN	LMAX	LMIN	LMAX
DVQM=0	RGBY	0	63	0	255	0	1023	0	4095
	CbCr	0	63	0	255	0	1023	0	4095
DVQM=1	RGBY	1	62	1	254	4	1019	16	4079
	CbCr	1	62	1	254	4	1019	16	4079
DVQM=2	RGBY	4	59	16	235	64	940	256	3760
	CbCr	4	60	16	240	64	960	256	3840

Computer formats (e.g. VESA DMT and CVT standards) normally set DVQM to 0, which causes the output signal to use the full gamut available for the given number of component bits. Television formats, on the other hand, typically set DVQM to 2 for a reduced gamut as is required by various television standards (e.g. section 5 of EIA/CEA-861-C).

In the television industry, a reduced code range is used to transmit nominal video content so as to leave room at the top and the bottom for overshoot, undershoot, and embedded timing references. Note that DVQM can be set to 1 in order to test undershoot & overshoot margins per section 7.11 of SMPTE 296M-2001 and section 5.3 of EIA/CEA-861-C. Since HDMI has a separate means for transmitting sync and data, the embedded timing references found in legacy television encodings are never used. Therefore, the extreme codes (reserved for this information in the legacy television standard encodings) are left unused in HDMI.

Gamma Correction (GAMC & GAMA)

Gamma correction can be optionally added to a signal when necessary. Gamma correction is enabled and disabled by setting GAMC to the values 1 and 0, respectively. Whenever gamma is enabled, the exponent used is controlled by the parameter GAMA as shown in Table 4. Gamma correction is preformed on normalized floating-point levels, which vary from zero to one.

Table 4 - Gamma Correction Logic

GAMC	DVST	L (normalized level)	L' (normalized & gamma corrected level)
0	X	X	L
1	13	0 = L < 0.0228	4.0*L
		0.0228 = L = 1.0	(1.1115*(L^(1/GAMA)))-0.1115
	Any Other	0 = L = 0.018	4.5*L
		0.018 < L = 1.0	(1.099 *(L^(1/GAMA)))-0.099

[1] Throughout this document, items shown in gray are either future features that are not yet supported or features that do not exist in the context of HDMI.

Double-Clocking (NCPP)

Sometimes, industry standard formats (e.g. 480i) have pixel rates that are below the minimum pixel rate restriction (of 25 MHz) required by the TMDS interface. In these cases, a feature known as "double-clocking" is used to raise the TMDS clock rate to an acceptable frequency.

Double-clocking is controlled by the `NCPP` format parameter. When `NCPP` is set to 1 (normal single-clocked), each pixel, output by the generator, is paired with one clock pulse. In this case, the pixel rate is equal to the TMDS clock rate. By setting `NCPP` to 2 (double-clocked), two clocks are send during each pixel period and the TMDS clock rate is raised to 2 times the pixel rate. For example, the 480i29 format timing has a pixel rate of 13.5 MHz, which is below the TMDS 25MHz limit. By setting NCPP to 2, an alternate format 480i2x29 is created that boosts the TMDS clock rate to 27MHz – a frequency that meets TMDS clock rate criteria.

EIA/CEA formats 6, 7, 8, 9, 21, 22, 23, 24, 44, 45, 50, 51, 54, 55, 58, and 59 are double-clocked. Double-clocked formats have the number of clocks per pixel parameter `NCPP` set to 2.

Double-clocked formats have the letters "2x" in their names. However, this is not to say that all formats, with "2x" in their name, are double clocked. Several progressive "2x" formats exist, having 480 (or more) active lines, that are single clocked with double the horizontal resolution instead (i.e. have the number of clocks per pixel parameter NCPP set to 1 and HTOT, HRES, HSPD, and HSPW parameters set to twice their usual values). EIA/CEA formats 14, 15, 29, and 30 are instances this alternate type of "2x" format. With these formats, NPPP may be set to either 1 or 2. Normal horizontal resolution is provided, when NPPP is set to 2 (see "Pixel Repetition" discussion below).

Pixel Repetition (NPPP)

The EIA/CEA-861-C standard defines a number of progressively-scanned formats, which support variable horizontal resolution. These formats maintain a fixed 1440 or 2880-pixel format timing (`HRES`) and use pixel repetition to provide different effective horizontal resolutions.

Because the format timing is fixed as pixel repetition is applied, the underlying hardware of the generator is not used support pixel repetition. Instead, special image rendering code is used. This code repeats pixels (while rending) according to the setting of an additional format parameter `NPPP`. By implementing pixel repetition in this fashion, horizontal total (`HTOT`) does not have to be constrained to a multiple of `NPPP`. Instead, `HRES` (timing) can remain set to the EIA/CEA-861-C standard value of 1440 or 2880-pixels as `NPPP` is varied to change the effective horizontal resolution.

All of the variable horizontal resolution formats, in our format library, have pixel repetition disabled (`NPPP` = 0) by default, meaning that all 1440 or 2880 clocked pixels, of these formats, are unique and displayed by default.

There are a number of "gaming" formats, which all have the letters "4x" in their name. EIA/CEA formats 10, 11, 12, 13, 25, 26, 27, and 28 are gaming formats. These formats keep HRES fixed at 2880-pixels and allow pixel-repetition to be varied over a 10-to-1 range – thereby providing effective resolutions of 288, 320, 360, 411, 480, 576, 720, 960, 1440, and 2880 pixels, respectively. In addition to the 10-to-1 range, gaming formats typically utilize a special "160/n" blanking scheme, which further reduces the number of active pixels to 256, 284, 320, 366, 427, 512, 640, 853, 1280, and 2560 pixels, respectively. The special blanking provides a horizontal safe area that insures that all of the pixels in a game will be visible on over scanned displays.

Not all formats, with letters "4x" in their name, are gaming formats. Formats 35, 36, 37, and 38 are not gaming formats. These formats allow pixel-repetition to be set to 4, 2, or 1 – thereby providing effective resolutions of 720, 1440, or 2880, respectively. These formats do not include the "160/n" bars typically found with the gaming formats mentioned above.

Not all formats, with letters "2x" in their name, are double-clocked. Formats 14, 15, 29, and 30 are not double-clocked. These formats allow pixel-repetition to be set to 2 or 1– thereby providing effective resolutions of 720 or 1440, respectively. These formats do not include the "160/n" bars typically found with the gaming formats mentioned above.

Once a variable horizontal resolution format has been selected, a special image called "PixelRep" can be subsequently used to conduct pixel repetition testing. When a variable horizontal resolution format is present along with the "PixelRep" image, the IMAGE STEP key can be pressed and the pixel repetition factor (`NPPP`=n) varied over a 1-to-2, 1-2-4, or 1-to-10 range via the IMAGE knob. In the case of a gaming format (10, 11, 12, 13, 25, 26, 27, or 28), as long as the IMAGE STEP key is activated (lit), "160/n" bars will appear to the left and right of the default 2880-pixel "PixelRep" image area, thereby reducing the horizontal active to approximately 2560 clocked pixels - as defined in the EIA/CEA-861-C standard. If `HRES` is adjusted away from 2880 pixels (which would yield a non-standard timing), the left and right bars will remain fixed at approximately 5.555…% of `HRES` each, for a total of 11.111…% additional horizontal blanking.

Note that double-clocking and pixel-repetition cannot be applied at the same time (see "Pixel Decimation Info" of AVI InfoFrame section for details as to why this is so). This is not a problem, however, since the "4x" formats have plenty of pixels to keep the TMDS pixel rate well above the minimum rate of 25MHz.

Data Packet Generator (DPG)

Just below the digital video generator (in **Figure 1**) is the data packet generator (DPG). This block collects and packetizes data from four sources (which will be discussed next): audio samples from the sound data mixer (SDM) of the DAG, clock recovery data from the audio clock capture unit (ACCU) of the DAG, InfoFrames from the IFG, and AVMUTE commands from the GCU. Once collected and packetized, by the DPG, data packets may be optionally sent to the Tx for transmission (along with video, sync, control, & display enable) to the display (sink) via TMDS physical interface.

Two DPG parameters control the presence and frequency of data packet types. The data packet type gate parameter `DPTG` is a bit mask that enables and disables the transmission of generic, audio sample, ACR, and general control packets. The data packet type repeat parameter `DPTR` is another bit mask that establishes the frequency of generic and general control packets. When gated-on by the `DPTG` parameter, the generic and general control packets can be programmed to occur every frame or just once. The IFG also has two similar controls, `IFTG` and `IFTR`, which control the presence and frequency of individual InfoFrame packet types, respectively. These are discussed later under the heading "InfoFrame-based Display-mode Automation".

As with most generator parameters, changing the parameters of the DPG does not have an immediate effect. Instead, the current settings of the DPTR & DPTG parameters do not take effect until an DPGU command is issued. Issuing an DPGU command causes the DPG to update the packet types it is sending, output a general control repetition once or repeatedly, and to begin or continue sending audio sample and ACR data packets if enabled.

Digital Audio Generator (DAG)

At the very bottom (of **Figure 1**) is the digital audio generator DAG, which delivers both audio samples and audio clock regeneration (ACR) information to the DPG. The DAG can provide audio samples from two different audio sources: an internal sonic (sine wave) data generator

(SDG) or an external SPDIF audio input. When an external source is used, it is attached via the connector marked "AV " on the generator.

Sound Data Mixer (SDM)

A sonic data mixer (SDM) is used to switch between the internal-SDG and external-SPDIF audio sources under the control of the `SDMG` command/query. Once set, data from the selected source is routed to the DPG.

Sound Data Generator (SDG)

The SDG is capable of generating sine waves with different rates and amplitudes. Rates and amplitudes are controlled by the `SRAT` and `SAMP` command/queries, respectively. Two additional informational queries: `SMIN?` and `SMAX?`, are available, which return the minima and maxima of the sine wave data for the current amplitude setting, respectively. The test images "Audio-L", "Audio-LF", "Audio-R", "Audio-RF", "Audio-LR", and "AudioLRF" can also be used to vary rates and amplitudes, as well as, gate content onto channels.

Audio Clock Capture Unit (ACCU)

The DAG has an ACCU, which outputs ACR data, consisting of the values N and CTS, to the DPG for transmission to the display (sink). The display (sink) needs this information in order to reconstruct an audio reference clock from the incoming TMDS clock according to the relation:

Equation 1

Farc = 128 * `ARAT` = (`PRAT` * `NCPP` * N) / CTS

Where Farc is the frequency of the recovered audio reference clock, `ARAT` is the audio sampling rate, `PRAT` is the pixel rate, `NCPP` is the number of clocks per pixel, N is a constant, and CTS is the cycle time stamp value. The ACCU receives configuration information from the `XACR` parameter cluster[2], which contains both N and CTS values. N and CTS values are automatically calculated and loaded into the `XACR` cluster, based on the current values of the format parameters `PRAT`, `NCPP`, and `ARAT`, whenever a `FMTU` (or `ALLU`) is executed. There is an `XACR` command/query, which can be used in combination with the `DPGU` command to manually change the value of N (or both N and CTS values) after `FMTU` has finished executing, but its use is not recommended. In cases where the audio reference clock (128 * `ARAT`) is asynchronous to the TMDS rate (`PRAT` * `NCPP`), remaining counts must be swallowed and the value of CTS must be allowed to continually jog between different values under the control of special hardware.

[2] The parameter naming convention and command language of the generator has been extended to include what we call "parameter clusters". These parameters are compound and contain two or more sub-parameters. In this document, the constituents of parameter clusters are referred to by a special colon-based syntax. In this syntax, the parameter cluster name precedes the name of one of its sub-parameters, separated by a colon (e.g. "XACR:N"). When programming the generator, the constituents of parameter clusters can queried and set using standard IEEE-488 SCPI "command tree" syntax. See http://www.scpiconsortium.org for details.

Audio/Video Mute (AVMUTE)

AVMUTE eliminates spurious pops or noises in the audio and flashes in the video during format changes. In combination with the new HDCP 1.1 specification, it also allows content protection algorithms to be frozen during, and immediately resumed after, a format change. Therefore, lengthy disruptions of protected content, that would otherwise be necessary to perform a complete re-authentication, are avoided.

AVMUTE information is transmitted to the display (sink) via general control packets, which were discussed under the heading "Data Packet Generator" above.

The generator currently supports AVMUTE whenever the HDMI output protocol is being used and a format change is made. SET_AVMUTE and CLEAR_AVMUTE messages are sent before and after each format change, respectively. In this way, simply switching formats can test basic AVMUTE functionality.

In the future, generators will be able to evaluate HDCP AVMUTE behavior. To do this, a new feature will be added to the generator firmware that will allow the "HdcpProd" test image PASS-count to be halted before and resumed after each format change - without a re-authentication being initiated. Currently, the generator does not have this capability.

The current state of AVMUTE can be obtained using the `MUTE?` query. The `MUTE?` query will return the values 0 and 1, when video and audio are unmuted and muted, respectively.

AVMUTE can be enabled and disabled using either the `MUTE` command or by directly setting the appropriate subparameters in the general control packet parameter cluster `XGCP` and parameters in of the data packet generator. Using the `MUTE` command is the simplest way to control the state of AVMUTE and does not require any knowledge of the XGCP parameter cluster or the DPG. AVMUTE can be simply enabled and disabled by sending the "`MUTE 1`" and "`MUTE 0`" commands, respectively.

The more complicated way of controlling the state of AVMUTE is to use a combination of GCU and DPG commands. In order for the SET_AVMUTE and CLEAR_AVMUTE messages to be communicated, bit-1 of both `DPTG` and `DPTR` parameters of the data packet generator should be set to one and used. This tells the DPG to send general control packets to the display repeatedly. The SET_AVMUTE and CLEAR_AVMUTE messages may then be sent to the display by executing the commands "`XGCP 1 0`" and "`XGCP 0 1`", respectively.

For example, to manually enable AVMUTE, the following commands might be sent:

```
DPTG 7    // enable general control packet (1 ORed w/existing)
DPTR 1    // send general control packet repeatedly (1 ORed w/existing)
XGCP 1 0  // with "AVMUTE set flag" true and "AVMUTE clear flag" false
DPGU      // update the DPG to use these new settings now
```

Once enabled, the state of AVMUTE can be verified using the `MUTE?` query. At this point, executing the command "`MUTE?`" would return the value "`1`" – indicating that AVMUTE is enabled.

Likewise, the following commands could be sent to manually disable AVMUTE:

```
DPTG 7    // enable general control packet (1 ORed w/existing)
DPTR 1    // send general control packet repeatedly (1 ORed w/existing)
XGCP 0 1  // with "AVMUTE set flag" false and "AVMUTE clear flag" true
DPGU      // update the DPG to use these new settings now
```

The state of AVMUTE can be verified using the MUTE? query once again. Executing the command "MUTE?" would now return the value "0" – indicating that the AVMUTE is disabled.

MUTE is a hardware command and MUTE? is a hardware status query. There is no saved parameter involved. Therefore, there may be times when the command value sent may not match the query value returned. This only means that the generator hardware has not yet had time to react to the command value requested.

InfoFrame-based Display-mode Automation

InfoFrames allow the generator to keep a display informed as to "what is coming down the pipe" so that it can present it optimally - without any viewer intervention. Previous systems would require viewers to constantly fiddle with their remote controls in order to put their displays into appropriate "modes" for given types of signals and content. With the new system, a frame-by-frame overview of what exactly is being transmitted can now be sent to the display, thereby allowing it to select appropriate display modes automatically.

There are different types of InfoFrames. Each type carries information regarding a different aspect of the audio/video transmission. The generator supports a total of six different types of InfoFrame. These include: auxiliary video information (AVI), generic vendor specific (GIF-A and GIF-B), source product description (SPD), MPEG information (MPG), and audio information (AUD).

As you will recall from **Figure 1**, the instrument has an InfoFrame generator (IFG) block, which is responsible for outputting InfoFrames. The InfoFrame Data output by this block is controlled by a set of parameter clusters[2]: XAVI, XGIF, XSPD, XSPD, XMPG, and XAUD, which are used to control the information content of the InfoFrames that are generated by the IFG.

A unique and powerful feature of Quantum Data's HDMI generator is its ability to automatically compile and load these parameter clusters - whenever a format, image, or (in some cases) image version change is detected. In this way, InfoFrame data is always guaranteed to be in sync with what is being output. The following sections describe how automatic InfoFrame data compilation works and what values one should expect under various conditions.

Parameters can also be manually changed via serial commands, if necessary, between IFGU, FMTU, or ALLU invocations. This feature might be used, for example, to test a display's response to invalid data.

There is an IFTG parameter, which is a bit mask that can be used to select which types of InfoFrames are enabled and disabled.

When enabled, each type of InfoFrame can be sent either once or repeatedly. The IFTR parameter is another bit mask that controls which InfoFrame types are repeated and which ones are not.

As with most generator parameters, changing the parameters of the IFG does not have an immediate effect. Instead, the current settings of these parameters do not take effect until an IFGU, FMTU, or ALLU command is issued. Any of these commands will cause the IFG to update the information that it is sending, output InfoFrame types that have not been set for repetition once, and begin repeating InfoFrames that have been slated for repetition.

When manually setting InfoFrame parameters, it is important that the FMTU and ALLU commands not be used. This is because invoking the FMTU and ALLU commands will cause all manual

InfoFrame settings to be overwritten with automatically generated values. Instead, only the IFGU command should be used after making manual changes.

The “PacketTx” image can be used to display the current InfoFrame values that are being transmitted. When manually controlling values in combination with the IFGU command and displaying the “PacketTx” image, one should follow the IFGU command with an IMGU command in order to refresh the “PacketTx” image.

AVI Overview

AVI InfoFrames contain the information that is necessary for display-mode automation. This information is automatically compiled from current format and image parameters, whenever format or image changes are made (e.g. when IFGU, FMTU, and ALLU commands are issued).

AVI compilation is a difficult subject, which is complicated by the fact that the EIA/CEA-861-C standard leaves some things open to interpretation. Table 5 shows the overall relationship between generator parameters and AVI InfoFrame parameters. This table refers to other tables, which explain the compilation process in finer detail (see Table 13, Table 16, and Table 17).

Table 5 lists Quantum Data (QDI) format control parameters on the left, related EIA/CEA AVI information parameters on the right, and relationships between the two in the middle. The table is broken into four sections – each section covering a different aspect of the display system.

The first section lists content mapping info. Here, we see five high-level generator format controls on the left and ten low-level AVI information parameters on the right. The parameters on the left control the logical aspect ratio of both the signal interface and the “active” image content flowing through it. They also drive the compilation of a detailed implementation as documented via the AVI-InfoFrame parameters on the right.

The second section of Table 5 lists the digital video signal type controls and information.

A third section of Table 5concerns the pixel repetition factor that is of some importance in connection with “2x” double-clocked and ”4x” gaming formats.

Finally, the fourth section of Table 5 contains an identification code that can be used to identify the current video format, in cases where it is one of the standard 59 formats listed in the EIA/CEA-861-C specification.

Table 5 - QDI-Format to AVI-InfoFrame Parameter Relationships

QDI-Format Parameter	Format Parameter Description	Relationship	AVI-InfoFrame Parameter	AVI-InfoFrame Parameter Description
SXAR	Signal aspect ratio	See Table 13	A	Active Format Information Present (flag)
			R	Active Format Aspect Ratio (ADF value)
SXEX	Signal-from-extended map		S	Scan Information (safe area information)
			SC	Non-uniform Picture Scaling (morph mode)
EXAR	Extended "shoot" aspect ratio		M	Picture Aspect Ratio (as "delivered")
			B	Bar Info (present flags)
EXCX	Extended-from-content map		ETB	Line number of end of top bar
			SBB	Line number of start of bottom bar
CXAR	Content aspect ratio		ELB	Pixel number of end of left bar
			SRB	Pixel number of start of right bar
DVSM	Digital video sampling method	See Table 16	Y	Color Encoding & Sampling Methods
DVST	Digital video signal type		C	Colorimetry
NCPP	Number of clocks per pixel	See Table 17	PR	Double Clocking & Pixel Repetition
NPPP	Number of pixels per virtual pixel			
DVIC	Digital video identification code	Equal	VIC	Video identification codes

Content Mapping Info (XAVI:A, R, S, SC, M, B, ETB, SBB, ELB, & SRB)

Consumer television and home theatre displays must often accept program material (content), which is not specifically formatted for the signal being received or the physical screen at hand. In some cases, the shape (aspect ratio) of the content may not match the shape of the signal's coded frame, picture-within-picture, or the display's physical screen. In other cases, the source may place image content very near an edge of the signal's coded frame so that important information (e.g. a portion of a lotto number) might be obscured if over-scanned by the display. A source might even morph content in order to fully utilize available signal bandwidth or eliminate content fitting bars – should they be deemed "annoying".

Given all of this variability, you might ask, "What is a display to do?" Today's intelligent displays can do allot of things to present content optimally, but they must first know what exactly they are receiving. This is where the content mapping info portion of AVI InfoFrame comes into play. The content mapping info contains content aspect ratio, active format description (AFD), safe area (over/under scan), bar, and morph status. Embedded within the AFD status are aspect ratio mapping, position, matting, shoot, and protect information. The generator is capable of automatically compiling and generating content mapping info from a single high-level model of the display system consisting of a simple set of aperture and mapping parameters. These are shown in Figure 2.

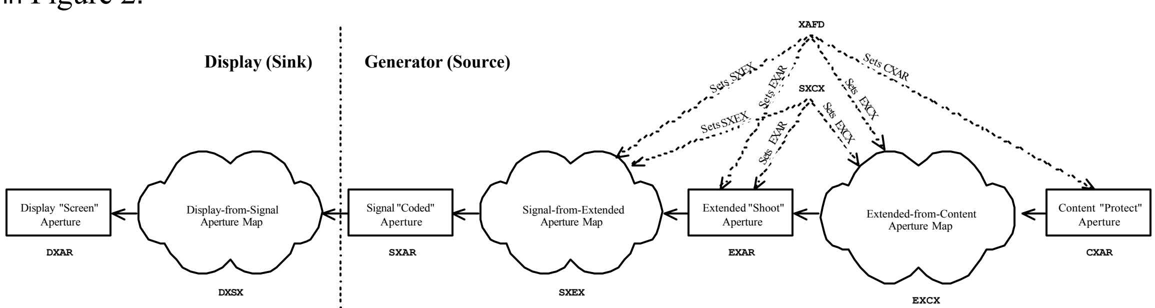


Figure 2 - High-level Aperture & Aperture-Mapping Parameters

AFD, Safe Area, and Morph Controls

The five generator parameters, shown on the lower right, are part of a format so they are selected, along with the timing, via the format knob. Three aspect ratios (CXAR, EXAR, and SXAR) and two aperture maps (SXEX and EXCX) have been defined for control purposes. When first loaded from a saved format, some of these parameters are set to default values. These values may be later modified in order to conduct AFD testing, by selecting a special "AFDtest" image and using the image step feature.

Aspect Ratios

Television signal interfaces and display screens typically have one of two different aspect ratios: 4x3 and 16x9. Computer signal interfaces and display screens are usually 4x3, but sometimes 5x4, 3x2, or 16x10 ratios are used. Modern motion picture film projectors typically support one (or more) of the following six ratios: ˜4x3, 13x9, 5x3, ˜13x7, ˜11x5, and (or) ˜12x5.

The shape of content varies as well. Most television programs continue to be shot at 4:3 (1.333…), while a growing number are shot at 16:9 (1.777…) - in order to accommodate the wider aspect of the new HDTV sets. Sometimes television content is shot in a compromise 14:9 "shoot & protect" format that lies midway between the 4:3 and 16:9 ratios. Here, a 16:9 image is "shot" with a "protected" 14:9 portion that can be more easily fitted into both 4:3 and 16:9 displays – without a lot of distracting waste. The shape of film content is quite variable. Old 35mm "Academy" classics have a ˜4x3 (˜1.37:1) aperture, which almost matches the 4x3 shape of standard definition video interfaces and display screens. Modern films, on the other hand, are usually shot with very wide ratios – usually wider than HDTV. They are typically composed & projected from a ˜13x7 (˜1.85) soft or hard matted[3] letterbox centered within the original 35mm classic Academy aperture - using flat lenses[4] on both ends of the process. In Western Europe, things are much the same, except that a 5x3 letterbox and screen are used. Sometimes (x2) anamorphic lenses are used to reshape content, on both ends of the process, so that the entire 35mm "Academy" aperture is utilized. In this case, one lens - in the camera - horizontally squeezes a ˜12x5 (˜2.39:1) scene into the 35mm Academy aperture, while a complimentary lens in the projector horizontally stretches the content back to normal for projection on a ˜12x5 (˜2.39:1) screen. 70mm films are normally shot and displayed with an ˜11x5 (˜ 2.20:1) aspect ratio – using a flat lenses on both ends. One of the widest films ever shot was MGM's "BEN-HUR", which had a vast 2.76:1 aspect ratio. Here, a 70mm process was used, but with (x1.25) anamorphic lenses on both ends. The IMAX process - a variation of flat 70mm - rotates and enlarges (by 3x) the image on the film so that the projected quality is significantly increased. IMAX captures and projects a 13x9 (˜1.44:1) image - again using a flat lens on both ends.

PXAR is the aspect ratio of pixels in active regions of the raster and matches the values listed in Tables 3 of EIA/CEA-861-C. PXAR is always set as follows:

PXAR = ((1+XAVI:PR)*(ratio indicated by value of the XAVI:M field)) / (HRES/VRES)

---

[3] Many motion pictures are composed on film having a 1.37:1 "Academy" ratio. In order to obtain the wide aperture of theatrical display, the top and bottom-most areas of the native "Academy" aperture are simply cropped or "matted". This matting operation can be done either sooner (e.g. in the camera), later (e.g. in the projector), somewhere in-between (e.g. in a DVD mastering process), or not at all. Films that are cropped immediately are said to be "hard" or "closed" matted, whereas films that are not, are said to be a "soft" or "open" matted. Maintaining a "soft" or "open" matte, throughout the production process, allows the same 1.37:1 content to be matted in a variety of ways – in the back end - in order to accommodate a variety of aspect ratios – such as those that would be more conducive to television and DVD distribution. Hard matting, on the other hand, insures that spurious content (e.g. boom microphones or naked body parts) will never be (accidentally) revealed in the final presentation.

[4] Two different types of lenses are used in film production and display: flat and anamorphic. Flat lenses do not distort the image, while anamorphic lenses shrink and stretch the image along the horizontal axis (usually by a factor of 2).

Automatic AVI Parameter Compilation

A high-level set of format parameters control the settings of the AVI InfoFrame parameters A, R, S, SC, M, B, ETB, SBB, ELB, and SRB in the automatic mode. This set consists of three aperture aspect ratios CXAR, EXAR, SXAR and two aperture-to-aperture maps EXCX and SXEX.

Both CXAR and EXAR represent the shape of program content. Modern television directors must keep either one or two of display screen shapes in mind, when shooting program content. Shooting content for display on a single screen shape is easy and only requires a single content ratio. In this case, both CXAR and EXAR are set equal to the shape of the content. Alternatively, content intended for two different shaped screens could be shot using a technique known as "shoot and protect". Here, CXA R and EXAR are different: CXAR represents "protected" content, that will appear on all displays, while EXAR represents an extended “shot” containing both “protected” as well as extra nonessential imagery for wider or taller displays. The nonessential portion of the EXAR aperture may be masked-off on narrower or shorter displays.

SXAR always represents the natural aspect ratio of the video signal format (or coded frame) that transports images to the display. For example, the format 480p59 has a 4:3 “T” aperture with a SXAR=1.333…, while the format 1080i29 has a 16:9 “H” aperture with a SXAR=1.777…. In the case of non-established formats, square pixels are assumed, and the ratio HRES/VRES is used (e.g. a 1280x1152 format would have a SXAR=1.111…).

The two maps EXCX and SXEX fit the three apertures CXAR, EXAR, and SXAR inside one another (see Figure 2). First, the extended-from-content mapping parameter EXCX fits the content aperture CXAR into the extended aperture EXAR. Next, the second signal-from-extended mapping parameter SXEX fits the extended aperture EXAR into the signal aperture SXAR. Both parameters utilize a bit-mask to describe the mapping methods used. This mask is constructed using the definitions shown in Table 9.

In the simplest case, all the apertures match exactly, so the mapping operations essentially "do nothing".

QDI Content Mapping Notation

A notation has also been established, consisting of letter symbols (with subscripts), which can be used to express standard aspect ratios, as well as, content protection and fitting methods. These are summarized in Table 6 and Table 7.

Table 6 - Established Aspect Ratios (Shapes)

AR	Symbol	Description	Example(s)
0.750	OT or T_{90}	“Television” Portrait	3x4, 480x640, 600x800, 768x1024, 960x1280, 1200x1600
0.800	OG or G_{90}	Workstation Graphics Portrait	4x5, 1024x1280, 1280x1600
1.000	Q	Quadrate, “Square”	1x1, 512x512, 1024x1024
1.250	G	Workstation Graphics	5x4, 1280x1024, 1600x1280
1.333…	T (A)[3]	Silent Film / SD “Television”	4x3, 640x480, 800x600, 1024x768, 1280x960, 1600x1200
˜1.37	C (B)[3]	Academy[1] “Classic” w/Sound	˜ 4x3, 0.825x0.602 SMPTE RP40 35mm “C”
1.444…	I	IMAX™	13x9, IMAX™
1.500	V (T)[3]	Vista Vision™	3x2, 1152x768 Apple Computer, (uncropped) Vista Vision™
1.555…	M (V)[3]	Mid	14x9, AFD Shoot & Protect (half-way between 4x3 and 16x9)
1.600	D	16 Decimal	16x10, 1728x1080, 1280x800, see VESA CVT 1.0
1.666…	E	European Film, “1.66”	5x3, 1200x720, 1280x768, 1800x1080
1.750	Z	Old Film	7x4, Old Metro-Goldwyn-Mayer & Disney Films
1.777…	H	HDTV	16x9, 1280x720,1920x1080
˜1.85	A (F)[3]	Film Standard (USA)	˜13x7, 1280x692, 1920x1038, 0.825x0.446 SMPTE RP40 35mm “A”
2.000	U	Univisum™	2x1, 1280x640, 1920x960
˜2.20	F (M)[3]	“Flat”, MPEG “20x9”	˜11x5, 1280x582, 1920x874, 1.912x0.870 SMPTE RP91 70mm
˜2.39	B (C)[3]	Anamorphic Cinema, “2.35”[2]	˜12x5, 1280x536, 1920x804, 1.650x0.690 SMPTE RP40 35mm “B”

Notes:

1. In 1932, the shape of film was changed from 1.33:1 to 1.37:1 in order to better accommodate the new optical soundtrack that was added a few years earlier. This slightly wider shape is the true aspect ratio of “classic” film. The aperture is commonly referred to as “Academy” and was the shape of the vast majority of U.S. films produced until the 1950s.
2. Sometimes called “2.35”, which was the aspect ratio before it was changed to 2.39:1 in 1971 to keep splices from showing up in the projected image.
3. The letters in parenthesis are the letters that we used to use, before we updated our aspect ratio symbols for compatibility with existing film industry standards (i.e. the symbols defined in SMPTE RP40).

Table 7 - Established Aspect Ratio Mapping & Safe Area Methods

Symbol	Description	Example(s)
N	Natural (do nothing)	4:3 content into 480i29 signal or 16:9 content into 1080i29 signal
L_{csp}	Shoot & Protect	14:9 shoot & protect zone within 16:9 signal aperture
S	Squeeze into (anamorphise)	16:9 content squeezed into the 4:3 aperture of 480p59 signal
S_u	Stretch to fill (deanamorphise)	16:9 squeezed content from 480p59 signal stretched to fill 16:9 screen
L_c[1]	Shrink to Letterbox/Pillar (centered)	16:9 content letterbox in 480i29 signal or a 4:3 content pillar in 1080i29
L_t[1]	Shrink to Letterbox/Pillar (at top)	
L_u	Enlarge from Letterbox/Pillar	16:9 letterbox from 480i29 signal enlarged to fill a 16:9 screen
K[2]	Confine Content to Safe Area	576x384 safe title area within 720x480
K_u	Enlarge Content from Safe Area	Safe area from over scanned signal enlarged to fill a under scanned screen

Notes:

1. The L method inserts either a “Letterbox” or a “Pillar” depending on the relationship between the aspect ratios of the source and destination apertures. A “Letterbox” will be inserted if the source aperture is wider than that of the destination. On the other hand, a “Pillar” will be inserted if the source aperture is narrower than the destination aperture.
2. The K method can be used to symmetrically shrink content so as to provide safe action or safe title areas (e.g. per SMPTE RP 218-2002).

Standard AFD Content Mapping Cases

In most AFD cases, the EXCX map is a do nothing and the EXAR-shaped extended aperture matches the CXAR-shaped content exactly. In these cases, CXAR-shaped content is (in effect) directly fitted into the SXAR-shaped signal aperture in a single step - in accordance with the value of the SXEX mapping parameter.

However, in three AFD cases, two maps (SXEX and EXCX) and an additional “extended” aperture (EXAR) are required to fit content into the signal aperture. The EXCX map is used first to fit CXAR-shaped content into the extended aperture EXAR. Next, the SXEX map is used to map EXAR-shaped material, in the extended EXAR staging area, into the SXAR-shaped signal interface. For example, the standard AFD fitting operation T-L_{ccb}-H-L_{csp}-T (shown in Example 2 of

Table 14) first shoots & protects (EXCX=L_{csp}) source content (CXAR=T) within an extended aperture (EXAR=H), then letterboxes (SXEX=L_{cbb}) the extended aperture (EXAR=H) into the aperture of the signal format (SXAR=T). You will note that the notation used here traces the path from the physical display screen, back through the video signal interface, and finally, to the source of image content as one reads from left-to-right.

SXCX is a command that automatically sets the SXEX, EXAR, and EXCX parameters to values necessary to fit CXAR-shaped image content into the SXAR-shaped aperture of the signal interface-timing format.

XAFD is also a command that automatically sets SXCX, SXEX, EXAR, EXCX, CXAR parameters two values necessary to support a given AFD code (see Table 13). Note, that this command will not change the value of SXAR, which remains fixed by the current format. Therefore, the range of allowed AFD values are limited to those associated with the value of SXAR in the current format.

A special test image called "AFDtest" is available for evaluating how well a display supports AFD.

A similar DXSX method has also been defined for the display side, which fits the aperture of the incoming SXAR-shaped signal interface-timing format into the DXAR-shaped aperture of the physical display screen. These parameters are beyond the scope of this document.

AFD Control Parameter Command Set

PXAR? Pixel Aspect Ratio Query
Syntax: `PXAR?`

CXAR Content Aspect Ratio
0.75 to ˜2.39 (see Table 6)

EXAR Extended Aspect Ratio
0.75 to ˜2.39 (see Table 6)

SXAR Signal Aspect Ratio (of coded frame)
0.75 to ˜2.39 (see Table 6)

SXCX Signal-From-Content Aperture Map (sets EXAR=CXAR, SXEX=SXCX, & EXCX=0)
0 to 131071 (see Table 7)

SXEX Signal-From-Extended Aperture Map
0 to 131071 (see Table 7)

EXCX Extended-From-Content Aperture Map
0 to 131071 (see Table 7)

XAFD AFD Mode Setting
0 to 15 sets SXCX, SXEX, EXAR, EXCX, CXAR associated with a given AFD code (see Table 13). This command will not change the value of SXAR, which remains fixed by the current format. Therefore, the range of allowed AFD values are limited to those associated with the value of SXAR in the current format.

XLBW Arbitrary Left Border Width
0 to 65535 establishes the height of the top border.

XRBW Arbitrary Right Border Width
0 to 65535 establishes the height of the top border.

XTBH Arbitrary Top Border Height
0 to 65535 establishes the height of the top border.

XBBH Arbitrary Bottom Border Height
0 to 65535 establishes the height of the top border.

Table 8 - Aspect Ratio Command De-Rounding Logic

Parameter Value Entered	Exact Fraction Saved	Exact Value Saved	Symbol
1.33 ≤ Value Entered ≤ 1.34	4/3	1.333…	T (A) [1]
1.37 ≤ Value Entered ≤ 1.38	0.825/0.602	1.370431…	C (B) [1]
1.44 ≤ Value Entered ≤ 1.45	13/9	1.444…	I
1.55 ≤ Value Entered ≤ 1.56	14/9	1.555…	M (V) [1]
1.66 ≤ Value Entered ≤ 1.67	5/3	1.666…	E
1.77 ≤ Value Entered ≤ 1.78	16/9	1.777…	H
1.84 ≤ Value Entered ≤ 1.85	0.825/0.446	1.849775…	A (F) [1]
2.19 ≤ Value Entered ≤ 2.21	1.912/0.870	2.197701…	F (M) [1]
2.35 ≤ Value Entered ≤ 2.40	1.650/0.690 [2]	2.391304… [2]	B (C) [1]

Notes:

1. The letters in parenthesis are the letters that we used to use, before we updated our aspect ratio symbols for compatibility with existing film industry standards (i.e. the symbols defined in SMPTE RP40).

2. Modern digital cinema cameras use the value 64/27 or 2.370370… (e.g. Thomson Viper FilmStream™).

Table 9 - Mapping Codes

Method	Keep Safe Area							Letterbox / Pillar							Squeeze / Stretch			Decimal Code	Symbol
Parameter	K[5:0]							L[5:0]							S[2:0]				
Bit	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Cases	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	N_0
	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	16	N_1
	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	32	N_2
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	S
	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	8	L_{csp}
	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	264	L_{cbb}
	0	0	0	0	0	0	0	0	1	0	0	0	1	1	0	0	0	280	L_{tbb}
	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	34816	K_{tbb}

Table 10 - Squeeze/Stretch "S" Field Codes

Squeeze/Stretch Mode	S[2:0]			Symbol
	Undo	Non-Linear	Squeeze	
	Bit 2	Bit 1	Bit 0	
Disabled (scaling is uniform)	X	X	0	
Linear Squeeze (anamorphise)	0	0	1	S
Linear Stretch (deanamorphise)	1	0	1	S_u
Non-Linear Squeeze	0	1	1	S_n
Non-Linear Stretch (undo)	1	1	1	S_{nu}

Table 11 - Letterbox/Pillar/Shoot & Protect "L" Field Codes

Letterbox/Pillar Mode	L[6:0]							Symbol
	Undo	Bars	BarContents[1]		Position		Shrink	
	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	
Disabled	X	X	X	X	X[2]	X[2]	0	
Centered shrink w/black bars top & bottom	0	1	0	0	0	0	1	L_{cbb}
Centered shrink w/gray bars top & bottom	0	1	0	1	0	0	1	L_{cgb}
Centered shrink w/white bars top & bottom	0	1	1	0	0	0	1	L_{cwb}
Centered shrink w/custom fill top & bottom	0	1	1	1	0	0	1	L_{ccb}
Centered protected shrink w/shot surround	0	0	X	X	0	0	1	L_{csp}
Fill screen with enlargement from center (undo)	1	X	X	X	0	0	1	L_{cu}
Top shrink w/single black bar at bottom	0	1	0	0	0	1	1	L_{tbb}
Top shrink w/single gray bar at bottom	0	1	0	1	0	1	1	L_{tgb}
Top shrink w/white bar at bottom	0	1	1	0	0	1	1	L_{twb}
Top shrink w/custom fill at bottom	0	1	1	1	0	1	1	L_{tcb}
Top protected shrink w/shot surround	0	0	X	X	0	1	1	L_{tsp}
Fill screen with enlargement from top (undo)	1	X	X	X	0	1	1	L_{tu}
Reserved	X	X	X	X	1	X	X	

Notes:
In the case of pixel repetition (NPPP $\neq$ 0), the color of the (extended) bars, on the left and right sides of the image (if present), are filled in accordance with the setting of the "L" method's "Bar Contents" field – even when the letterbox/pillar fitting method is disabled (bit 3 = 0).

1. These bits are used to sequence redundant AFD codes.

Table 12 - (Keep) Safe Area “K” Field Codes

(Keep) Safe Area Mode	K[6:0]							Symbol
	Undo	Bars	BarContents		Source	Safe Area		
	Bit 16	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	
None (w/safe area markers)	X	X	X	X	0	0	0	
Shrink coded frame to action area w/opaque black bar surround	0	1	0	0	0	0	1	K_{abb}
Shrink coded frame to action area w/opaque gray bar surround	0	1	0	1	0	0	1	K_{agb}
Shrink coded frame to action area w/opaque white bar surround	0	1	1	0	0	0	1	K_{awb}
Shrink coded frame to action area w/opaque bar surround having custom fill	0	1	1	1	0	0	1	K_{acb}
Coded frame has shot-protected action area w/o graticules or bars	0	0	0	0	0	0	1	K_{asp}
Coded frame has shot-protected action area w/action graticule w/o bars	0	0	0	1	0	0	1	K_{aspa}
Coded frame has shot-protected action area w/title graticule w/o bars	0	0	1	0	0	0	1	K_{aspt}
Coded frame has shot-protected action area w/action & title graticules w/o bars	0	0	1	1	0	0	1	K_{aspb}
Fill screen with action area (undo)	1	X	X	X	0	0	1	K_{au}
Shrink coded frame to title area w/opaque black bar surround	0	1	0	0	0	1	0	K_{tbb}
Shrink coded frame to title area w/opaque gray bar surround	0	1	0	1	0	1	0	K_{tgb}
Shrink coded frame to title area w/opaque white bar surround	0	1	1	0	0	1	0	K_{twb}
Shrink coded frame to title area w/opaque bar surround having custom fill	0	1	1	1	0	1	0	K_{tcb}
Coded frame has shot-protected title area w/o graticules or bars	0	0	0	0	0	1	0	K_{tsp}
Coded frame has shot-protected title area w/action graticule w/o bars	0	0	0	1	0	1	0	K_{tspa}
Coded frame has shot-protected title area w/title graticule w/o bars	0	0	1	0	0	1	0	K_{tspt}
Coded frame has shot-protected title area w/action & title graticules w/o bars	0	0	1	1	0	1	0	K_{tspb}
Fill screen with title area (undo)	1	X	X	X	0	1	0	K_{tu}
Shrink coded frame to custom area w/opaque black custom border	0	1	0	0	0	1	1	K_{cbb}
Shrink coded frame to custom area w/opaque gray custom border	0	1	0	1	0	1	1	K_{cgb}
Shrink coded frame to custom area w/opaque white custom border	0	1	1	0	0	1	1	K_{cwb}
Shrink coded frame to custom area w/opaque custom border having custom fill	0	1	1	1	0	1	1	K_{ccb}
Coded frame has shot-protected custom area w/o graticules or border	0	0	0	0	0	1	1	K_{csp}
Coded frame has shot-protected custom area w/action graticule w/o border	0	0	0	1	0	1	1	K_{cspa}
Coded frame has shot-protected custom area w/title graticule w/o border	0	0	1	0	0	1	1	K_{cspt}
Coded frame has shot-protected custom area w/action & title graticules w/o border	0	0	1	1	0	1	1	K_{cspb}
Fill screen with custom area (undo)	1	X	X	X	0	1	1	K_{cu}

Table 13 - Established AFD-to-AVI InfoFrame Parameter Relationships

Case	Example	SXAR	SXEX[3,4]	EXAR	EXCX[3,4]	CXAR[3]	XAVI:M[5]	XAVI:R Or XAFD[9]	XAVI:A[6]	XAVI:B[7]	XAVI:SC[8]
1	1	T	N_0	CXAR	N_0	T	1	8	1	0	0
2	1[1]	T	N_1	CXAR	N_0	T	1	9	1	0	0
3	2	T	**L_{cbb}**	H	**L_{csp}**	T	1	15	1	2	0
4	3	T	L_{csp}	CXAR	N_0	M	1	13	1	0	0
5	4	T	L_{cbb}	CXAR	N_0	M	1	11	1	2	0
6	5	T	L_{tbb}	CXAR	N_0	M	1	3	1	2	0
7	6	T	**L_{cbb}**	H	**L_{csp}**	M	1	14	1	2	0
8	7	T	L_{cbb}	CXAR	N_0	H	1	10	1	2	0
9	8	T	L_{tbb}	CXAR	N_0	H	1	2	1	2	0
10	9[2]	T	L_{cbb}	CXAR	N_0	A	1	4	1	2	0
11	10[2]	T	L_{cbb}	CXAR	N_0	U	1	4	1	2	0
12	11[2]	T	L_{cbb}	CXAR	N_0	F	1	4	1	2	0
13	12[2]	T	L_{cbb}	CXAR	N_0	B	1	4	1	2	0
14	14	T	S-H-L_{csp}	CXAR	N_0	T	2	15	1	0	1
15	15	T	S-H-L_{cbb}	CXAR	N_0	T	2	9	1	1	1
16	16	T	S-H-L_{csp}	CXAR	N_0	M	2	14	1	0	1
17	17	T	S-H-L_{cbb}	CXAR	N_0	M	2	11	1	1	1
18	17[1]	T	S-H-L_{tbb}	CXAR	N_0	M	2	3	1	1	1
19	18	T	**S-H-L_{cbb}**	T	**L_{csp}**	M	2	13	1	1	1
20	19	T	S-H-N_0	CXAR	N_0	H	2	8	1	0	1
21	19[1]	T	S-H-N_1	CXAR	N_0	H	2	2	1	0	1
22	19[1]	T	S-H-N_2	CXAR	N_0	H	2	10	1	0	1
23	20[2]	T	S-H-L_{cbb}	CXAR	N_0	A	2	4	1	2	1
24	21[2]	T	S-H-L_{cbb}	CXAR	N_0	U	2	4	1	2	1
25	22[2]	T	S-H-L_{cbb}	CXAR	N_0	F	2	4	1	2	1
26	23[2]	T	S-H-L_{cbb}	CXAR	N_0	B	2	4	1	2	1
27	14	H	L_{csp}	CXAR	N_0	T	2	15	1	0	0
28	15	H	L_{cbb}	CXAR	N_0	T	2	9	1	1	0
29	16	H	L_{csp}	CXAR	N_0	M	2	14	1	0	0
30	17	H	L_{cbb}	CXAR	N_0	M	2	11	1	1	0
31	17[1]	H	L_{tbb}	CXAR	N_0	M	2	3	1	1	0
32	18	H	**L_{cbb}**	T	**L_{csp}**	M	2	13	1	1	0
33	19	H	N_0	CXAR	N_0	H	2	8	1	0	0
34	19[1]	H	N_1	CXAR	N_0	H	2	2	1	0	0
35	19[1]	H	N_2	CXAR	N_0	H	2	10	1	0	0
36	20[2]	H	L_{cbb}	CXAR	N_0	A	2	4	1	2	0
37	21[2]	H	L_{cbb}	CXAR	N_0	U	2	4	1	2	0
38	22[2]	H	L_{cbb}	CXAR	N_0	F	2	4	1	2	0
39	23[2]	H	L_{cbb}	CXAR	N_0	B	2	4	1	2	0

Notes:

1. Redundant cases are shown in gray.
2. In these cases, CXAR can actually be any ratio greater than 1.777… (16:9), while only the A, U, F, or B cases are enumerated here.
3. Compound maps, involving the EXAR extended aperture and both mapping methods SXEX & EXCX, are shown in bold-faced type. In these cases, an EXCX L_{csp}-opertion is first used to place CXAR-shaped content into the EXAR-shaped extended aperture. This operation is then followed by a SXEX L_{cbb}-operation, which finally places the EXAR-shaped extended aperture into the SXAR-shaped output signal aperture.
4. SXEX and EXCX may include an additional K_a or K_t safe area operation (e.g. $K_aL_{cbb}HL_{csp}$ or $K_tL_{cbb}HL_{csp}$) without affecting these relationships – except for XAVI:B (see note 6). If multiple safe area operations are specified, then the most restrictive area will override the rest in the order CX, followed by EX, followed by SX.
5. if(SXEX == S); then XAVI:M set to same aspect ratio as EXAR; else XAVI:M set to same aspect ratio as SXAR.
6. Set to zero for non-established cases.
7. if(SXEX or EXCX specifies Lcbb, Lcgb, Lcwb, or Lccb and content being fitted is wider than SXAR); then XAVI:B set to 2; else XAVI:B set to 1. if(SXCX, SXEX, or EXCX includes Kabb, Kagb, Kawb, Kacb, Ktbb, Ktgb, Ktwb, or Ktcb); then XAVI:B set to 3. Otherwise, XAVI:B is set to 0.

8. if(SXEX does not contain not S); then set XAVI:SC to 0; else if(content being fitted is wider the SXAR); then XAVI:SC set to 1; else XAVI:SC set to 2. When SXEX contains S, T-S-H mapping operation is assumed.

9. XAFD can be used to automatically set SXEX, EXAR, EXCX, and CXAR to the values shown in this table for a given XAVI:R value within the context of the SXAR of the current signal fomat. For example, if the format is 480i29 (which has an SXAR=1.333…) is present and the command “XAFD 15” is given, then SXEX, EXAR, EXCX, and CXAR will be set to L_{cbb}, 1.777…, L_{csp}, and 1.333…, respectively.

Table 14 – 4:3 “T” Shaped Screen AFD Examples

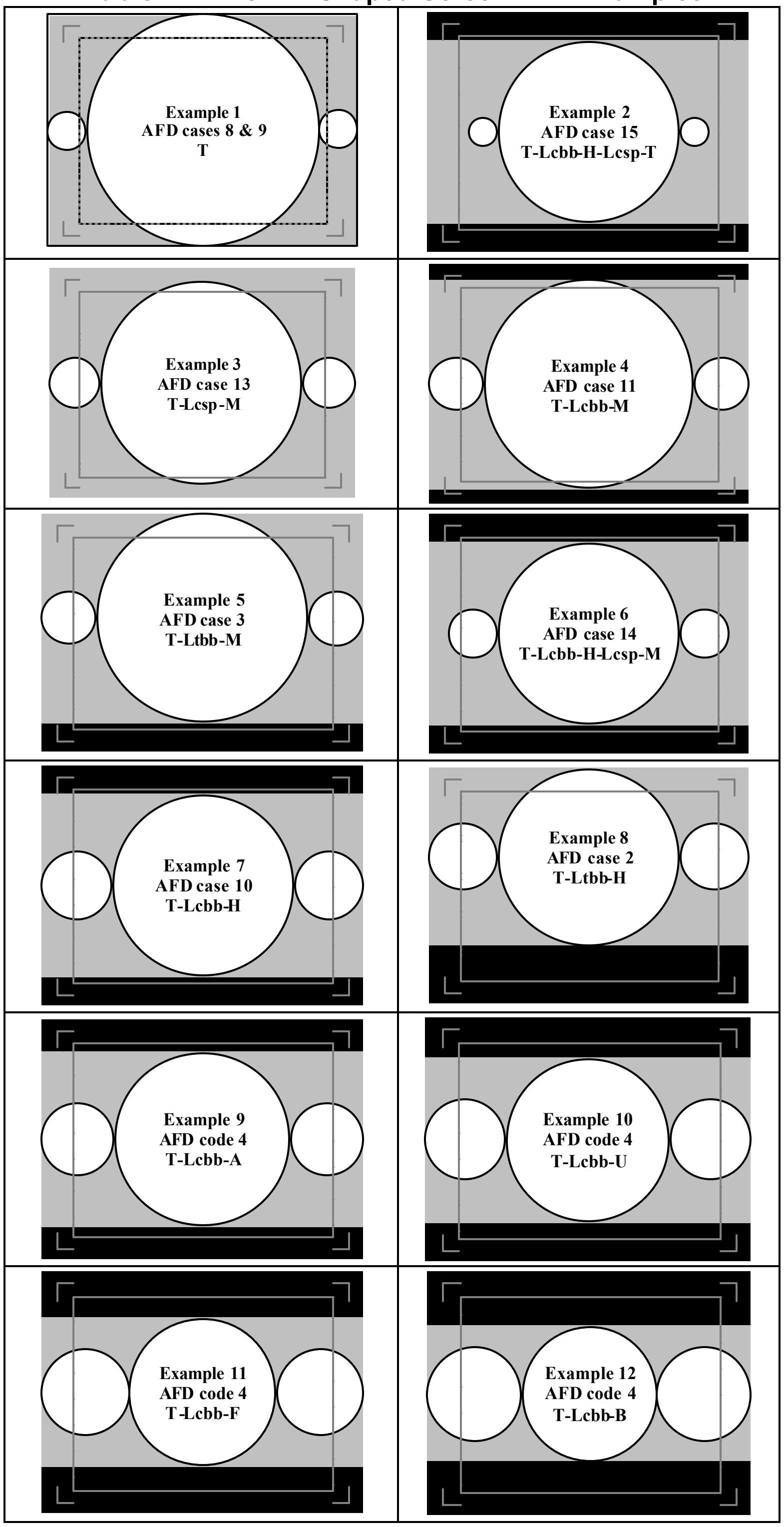
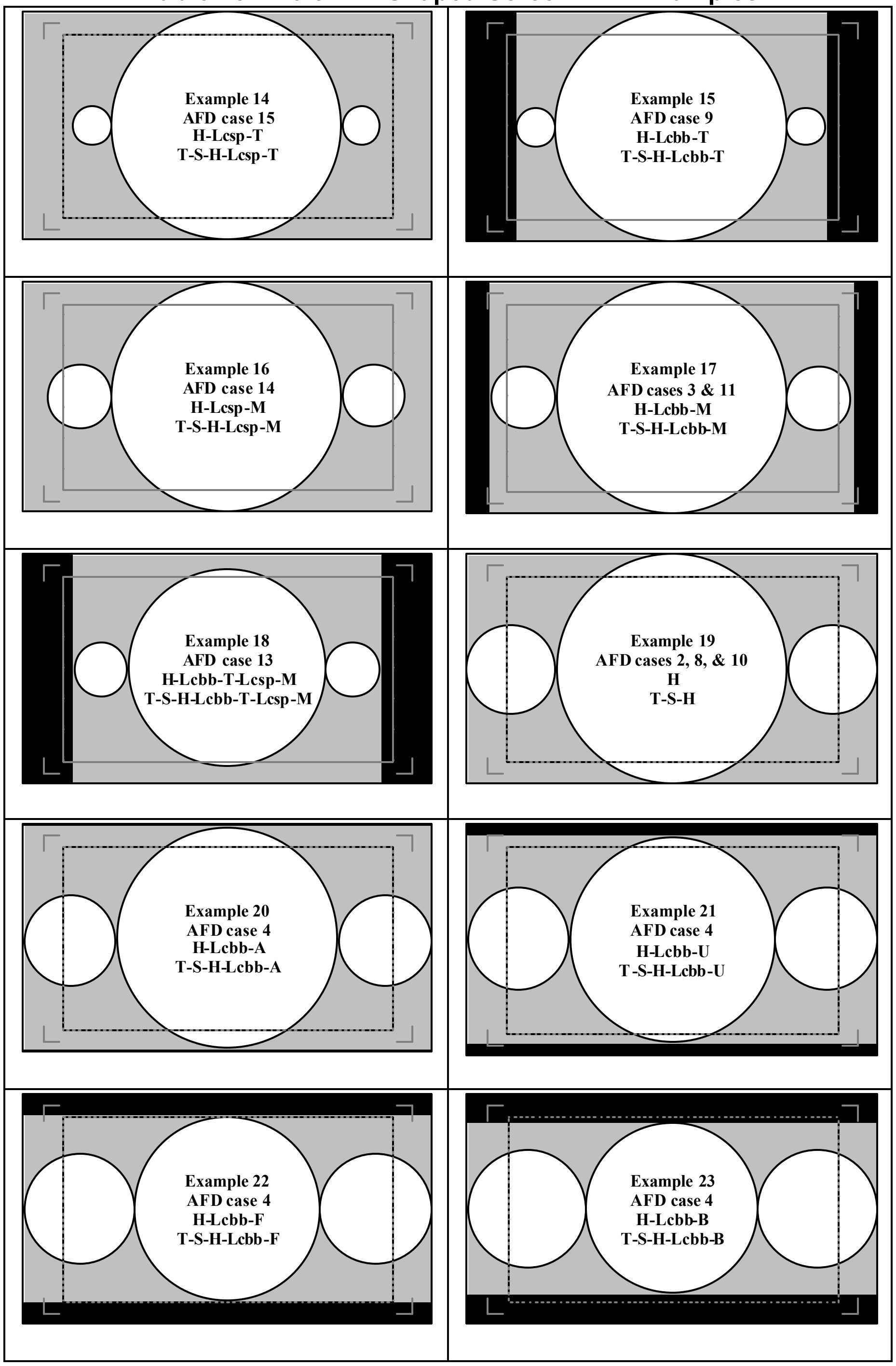


Table 15 – 16:9 "H" Shaped Screen AFD Examples



The Definition of Pixel Aspect Ratio & Setting of AVI:M

Section 2.4 of the EIA/CEA-861-C standard defines the term "picture aspect ratio" as "the ratio of width to height dimension of the picture as delivered across the uncompressed digital interface, including any top, bottom, or side bars". Tables 3, in section 4.1 of the EIA/CEA-861-C standard, lists both picture and pixel aspect ratios for various formats.

On the generator, the setting of XAVI:M depends on three format parameters: SXAR, EXAR, and SXEX. Where SXAR represent the "natural" aspect ratio of the video signal format (or "coded frame"), EXAR represents the aspect ratio of an "extended" aperture containing both essential and non-essential content - excluding any bars, and SXEX represents the mapping operation that is used to fit the EXAR content into the SXAR signal format.

There are cases where EXAR and SXAR are different and where a SXEX mapping operation must be used. In most of the cases, SXEX is a letterbox operation that fits EXAR content into the SXAR signal format using bars. Here, the "picture aspect ratio" remains equal to SXAR, because (by definition) the "picture aspect ratio" includes "any top, bottom, or side bars".

The EIA/CEA-861-C standard does not explicitly say how the "picture aspect ratio" should be defined in the cases where non-uniform scaling is used, however. To maintain the values in Table 3 of the standard, with non-uniform scaling, the "picture aspect ratio" must be defined such that the aspect ratio of the virtual undistorted EXAR content, being "delivered across" the SXAR-shaped video signal format, is represented by value of XAVI:M instead.

For example, in the case of 480p59SH anamorphic squeeze, the InfoFrame "picture aspect ratio" parameter XAVI:M is set to 2 and the "Non-uniform Picture Scaling" parameter XAVI:SC is set to 1. These values indicate that horizontally anamorphically squeezed 16:9 content is being "delivered across" the 4:3 coded frame of the 480p59 signal format, using a horizontal non-uniform picture scaling operation. In terms of generator parameters, the virtual undistorted EXAR image being transported is 16:9, the coded frame SXAR is 4:3, and the non-uniform scaling operation SXEX is S_h. This signal would appear distorted if displayed, on an SXAR-shaped display, without further processing. In order to obtain an undistorted picture, the display can sense the situation and apply additional processing since the 480p59 4:3 timing is 4:3 and the undistorted content (signified by XAVI:M==2) is 16:9.

To summarize, the value of M in the AVI InfoFrame is always set as follows:
if (SXEX == S)
then XAVI:M set to same aspect ratio as EXAR
else XAVI:M set to same aspect ratio as SXAR

Note that this means that, in the case of anamorphic squeeze or stretch, the EIA/CEA-861-C standard is currently limited to communicating only 4:3 and 16:9 content aspect ratios. This is because these are the only ratios supported by current XAVI:M InfoFrame parameter options. Other ratios (such as 13x7 “A”, which would be quite useful for advanced home theatre and digital cinema applications) could be supported if the EIA/CEA-861-C standard were suitably extended.

Colorimetry, Component Type, and Sampling Info (XAVI:Y, C)

Table 16 - Digital Video Type to AVI InfoFrame Parameter Relationships

DVST	DVSM	DVQM	GAMC	GAMA	XAVI:Y	XAVI:C	Description
10	0	0	0	X	0	0	RGB 4:4:4
13	2	2	1	2.222	1	0	YCbCr 4:2:2 SMPTE 260M-1999 Table 1
	4	2	1	2.222	2	0	YCbCr 4:4:4 SMPTE 260M-1999 Table 1
14	2	2	1	2.222	1	1	YCbCr 4:2:2 ITU-R BT.601-5 Table 3, Item 7
	4	2	1	2.222	2	1	YCbCr 4:4:4 ITU-R BT.601-5 Table 3, Item 7
15	2	2	1	2.222	1	2	YCbCr 4:2:2 ITU-R BT.709-5 Part 1, Section 6.10
	4	2	1	2.222	2	2	YCbCr 4:4:4 ITU-R BT.709-5 Part 1, Section 6.10

Pixel Decimation Info (XAVI:PR)

You will recall that there are two parameters: `NCPP` & `NPPP`, which separately account for the degree of pixel decimation that accompanies double-clocking and pixel-repetition operations, respectively. They are defined as follows:

`NCPP` Number of clocks per pixel (double-clocking factor applies to the whole line)
1 or 2

`NPPP` Number of pixels per pixel (repetition factor applies only to the active portion of line)
0 (disable repetition mode & remove extra left and right bars if present)
1 to 10 (enable pixel repetition mode & maybe insert extra left and right repetition bars if a gaming format is present)

Double-clocking and pixel-repetition cannot be applied at the same time, since there is only a single parameter (`XAVI:PR`), in the current AVI InfoFrame structure, to communicate pixel-decimation factor. Therefore, the value of these two parameters must insure that the condition ((`NCPP`-1)*`NPPP`) == 0 remains true at all times.

Section 6.4 (paragraph 8) of the EIA/CEA-861-C standard states that "In all cases, the first transmitted pixel data of a line of video is unique. Subsequent pixels will be repetition(s) of the previous pixel if pixel-repetition is used". So, in the cases where the clocked width of the active area (`HRES`) of the line is not an even multiple of (`NCPP`*`NPPP`), the image will be de-centered (or biased) to the left and the x-coordinates of all pixels in pixel-repetition test patterns will either be zero or a multiple of (`NCPP`*`NPPP`).

The state of "2x" double-clocking and "4x" pixel-repetition factors are reflected in the setting of the PR pixel-repetition field of the AVI InfoFrame `XAVI:PR`. The value of the PR field is a function of the values `NCPP` & `NPPP` and is set using the following relation:

if (`NPPP` == 0)
then `XAVI:PR`=(`NCPP`)-1
else `XAVI:PR`=(`NPPP`)-1

where: the condition ((`NCPP`-1)*`NPPP`) == 0 remains true at all times – this is checked

Table 17 - Pixel Repetition & Double-Clocking to AVI InfoFrame Parameter Relationships

NCPP	NPPP	XAVI:PR	XAVI:B	XAVI:ELB	XAVI:SRB	Description
1	0	0	X	X	X	Single-clocking with pixel repetition disabled
2	0	1	X	X	X	Double-clocking with pixel repetition disabled
1	1	0	1	160	2721	Single-clocking with zero repetition and side bars
1	2	1	1	160	2721	Single-clocking with single repetition and side bars
1	3	2	1	162	2722	Single-clocking with 2 repetitions and side bars
1	4	3	1	160	2721	Single-clocking with 3 repetitions and side bars
1	5	4	1	160	2721	Single-clocking with 4 repetitions and side bars
1	6	5	1	162	2725	Single-clocking with 5 repetitions and side bars
1	7	6	1	161	2724	Single-clocking with 6 repetitions and side bars
1	8	7	1	160	2721	Single-clocking with 7 repetitions and side bars
1	9	8	1	162	2719	Single-clocking with 8 repetitions and side bars
1	10	9	1	160	2721	Single-clocking with 9 repetitions and side bars

HDMI formats and the EIA/CEA-861-C Format Library

HDMI formats are television-oriented and taken directly from the EIA/CEA-861-C "A DTV Profile for Uncompressed High Speed Digital Interfaces" standard. EIA/CEA-861-C has a total of 59 numbered formats. Within this group, there are many slight variations, which bring the total to 412. Most of the variations involve the introduction of pixel repetition. Our test strategy is to test pixel repetition modes using a special test image, rather than expand our format library. When the pixel repetition format variations are removed, the number of remaining unique formats is reduced to 146 as listed in Table 18 below.

Table 18 - EIA/CEA-861-C Format Library

Case	DVIC	Format Name	NCPP[1]	SXAR[2]	SXEX[2]	CXAR[2]	TUNE[3]	HRES[4]	VTOT[5]
1	1	DMT0659	1	T	N_0	T	0.999000999	640	525
2	1	DMT0660	1	T	N_0	T	1.000000000	640	525
3	2	480p59	1	T	N_0	T	0.999000999	720	525
4	2	480p60	1	T	N_0	T	1.000000000	720	525
5	2	480p59LH	1	T	L_{cbb}	H	0.999000999	720	525
6	2	480p60LH	1	T	L_{cbb}	H	1.000000000	720	525
7	3	480p59SH	1	T	S	H	0.999000999	720	525
8	3	480p60SH	1	T	S	H	1.000000000	720	525
9	4	720p59	1	H	N_0	H	0.999000999	1280	750
10	4	720p60	1	H	N_0	H	1.000000000	1280	750
11	5	1080i29	1	H	N_0	H	0.999000999	1920	1125
12	5	1080i30	1	H	N_0	H	1.000000000	1920	1125
13	6	480i2x29	2	T	N_0	T	0.999000999	720	525
14	6	480i2x30	2	T	N_0	T	1.000000000	720	525
15	6	480i2xL1	2	T	L_{cbb}	H	0.999000999	720	525
16	6	480i2xL2	2	T	L_{cbb}	H	1.000000000	720	525
17	7	480i2xS1	2	T	S	H	0.999000999	720	525
18	7	480i2xS2	2	T	S	H	1.000000000	720	525
19	8	240p2x_1	2	T	N_0	T	0.999000999	720	262
20	8	240p2x_2	2	T	N_0	T	1.000000000	720	262
21	8	240p2x_3	2	T	N_0	T	0.999000999	720	263
22	8	240p2x_4	2	T	N_0	T	1.000000000	720	263
23	8	240p2xL1	2	T	L_{cbb}	H	0.999000999	720	262
24	8	240p2xL2	2	T	L_{cbb}	H	1.000000000	720	262
25	8	240p2xL3	2	T	L_{cbb}	H	0.999000999	720	263
26	8	240p2xL4	2	T	L_{cbb}	H	1.000000000	720	263
27	9	240p2xS1	2	T	S	H	0.999000999	720	262
28	9	240p2xS2	2	T	S	H	1.000000000	720	262
29	9	240p2xS3	2	T	S	H	0.999000999	720	263
30	9	240p2xS4	2	T	S	H	1.000000000	720	263
31	10	480i4x29	1	T	N_0	T	0.999000999	2880	525
32	10	480i4x30	1	T	N_0	T	1.000000000	2880	525
33	10	480i4xL1	1	T	L_{cbb}	H	0.999000999	2880	525
34	10	480i4xL2	1	T	L_{cbb}	H	1.000000000	2880	525
35	11	480i4xS1	1	T	S	H	0.999000999	2880	525
36	11	480i4xS2	1	T	S	H	1.000000000	2880	525
37	12	240p4x_1	1	T	N_0	T	0.999000999	2880	262
38	12	240p4x_2	1	T	N_0	T	1.000000000	2880	262
39	12	240p4x_3	1	T	N_0	T	0.999000999	2880	263
40	12	240p4x_4	1	T	N_0	T	1.000000000	2880	263
41	12	240p4xL1	1	T	L_{cbb}	H	0.999000999	2880	262
42	12	240p4xL2	1	T	L_{cbb}	H	1.000000000	2880	262
43	12	240p4xL3	1	T	L_{cbb}	H	0.999000999	2880	263
44	12	240p4xL4	1	T	L_{cbb}	H	1.000000000	2880	263
45	13	240p4xS1	1	T	S	H	0.999000999	2880	262
46	13	240p4xS2	1	T	S	H	1.000000000	2880	262
47	13	240p4xS3	1	T	S	H	0.999000999	2880	263
48	13	240p4xS4	1	T	S	H	1.000000000	2880	263
49	14	480p2x59	1	T	N_0	T	0.999000999	1440	525
50	14	480p2x60	1	T	N_0	T	1.000000000	1440	525
51	14	480p2xL1	1	T	L_{cbb}	H	0.999000999	1440	525

52	14	480p2xL2	1	T	L_{cbb}	H	1.000000000	1440	525
53	15	480p2xS1	1	T	S	H	0.999000999	1440	525
54	15	480p2xS2	1	T	S	H	1.000000000	1440	525
55	16	1080p59	1	H	N_0	H	0.999000999	1920	1125
56	16	1080p60	1	H	N_0	H	1.000000000	1920	1125
57	17	576p50	1	T	N_0	T	1.000000000	720	625
58	17	576p50LH	1	T	L_{cbb}	H	1.000000000	720	625
59	18	576p50SH	1	T	S	H	1.000000000	720	625
60	19	720p50	1	H	N_0	H	1.000000000	1280	750
61	20	1080i25	1	H	N_0	H	1.000000000	1920	1125
62	21	576i2x25	2	T	N_0	T	1.000000000	720	625
63	21	576i2xLH	2	T	L_{cbb}	H	1.000000000	720	625
64	22	576i2xSH	2	T	S	H	1.000000000	720	625
65	23	288p2x_1	2	T	N_0	T	1.000000000	720	312
66	23	288p2x_2	2	T	N_0	T	1.000000000	720	313
67	23	288p2x_3	2	T	N_0	T	1.000000000	720	314
68	23	288p2xL1	2	T	L_{cbb}	H	1.000000000	720	312
69	23	288p2xL2	2	T	L_{cbb}	H	1.000000000	720	313
70	23	288p2xL3	2	T	L_{cbb}	H	1.000000000	720	314
71	24	288p2xS1	2	T	S	H	1.000000000	720	312
72	24	288p2xS2	2	T	S	H	1.000000000	720	313
73	24	288p2xS3	2	T	S	H	1.000000000	720	314
74	25	576i4x25	1	T	N_0	T	1.000000000	2880	625
75	25	576i4xLH	1	T	L_{cbb}	H	1.000000000	2880	625
76	26	576i4xSH	1	T	S	H	1.000000000	2880	625
77	27	288p4x_1	1	T	N_0	T	1.000000000	2880	312
78	27	288p4x_2	1	T	N_0	T	1.000000000	2880	313
79	27	288p4x_3	1	T	N_0	T	1.000000000	2880	314
80	27	288p4xL1	1	T	L_{cbb}	H	1.000000000	2880	312
81	27	288p4xL2	1	T	L_{cbb}	H	1.000000000	2880	313
82	27	288p4xL3	1	T	L_{cbb}	H	1.000000000	2880	314
83	28	288p4xS1	1	T	S	H	1.000000000	2880	312
84	28	288p4xS2	1	T	S	H	1.000000000	2880	313
85	28	288p4xS3	1	T	S	H	1.000000000	2880	314
86	29	576p2x50	1	T	N_0	T	1.000000000	1440	625
87	29	576p2xLH	1	T	L_{cbb}	H	1.000000000	1440	625
88	30	576p2xSH	1	T	S	H	1.000000000	1440	625
89	31	1080p50	1	H	N_0	H	1.000000000	1920	1125
90	32	1080p23	1	H	N_0	H	0.999000999	1920	1125
91	32	1080p24	1	H	N_0	H	1.000000000	1920	1125
92	33	1080p25	1	H	N_0	H	1.000000000	1920	1125
93	34	1080p29	1	H	N_0	H	0.999000999	1920	1125
94	34	1080p30	1	H	N_0	H	1.000000000	1920	1125
95	35	480p4x59	1	T	N_0	T	0.999000999	2880	525
96	35	480p4xL1	1	T	L_{cbb}	H	0.999000999	2880	525
97	35	480p4x60	1	T	N_0	T	1.000000000	2880	525
98	35	480p4xL2	1	T	L_{cbb}	H	1.000000000	2880	525
99	36	480p4xS1	1	T	S	H	0.999000999	2880	525
100	36	480p4xS2	1	T	S	H	1.000000000	2880	525
101	37	576p4x50	1	T	N_0	T	1.000000000	2880	625
102	37	576p4xLH	1	T	L_{cbb}	H	1.000000000	2880	625
103	38	576p4xSH	1	T	S	H	1.000000000	2880	625
104	39	108Oi25_	1	H	N_0	H	1.000000000	1920	1250
105	40	1080i50	1	H	N_0	H	1.000000000	1920	1125
106	41	720p100	1	H	N_0	H	1.000000000	1280	750
107	42	576p100	1	T	N_0	T	1.000000000	720	625
108	42	576p100L	1	T	L_{cbb}	H	1.000000000	720	625
109	43	576p100S	1	T	S	H	1.000000000	720	625
110	44	576i2x50	2	T	N_0	T	1.000000000	720	625
111	44	576i2xL1	2	T	L_{cbb}	H	1.000000000	720	625
112	45	576i2xS1	2	T	S	H	1.000000000	720	625
113	46	1080i59	1	H	N_0	H	1.000000000	1920	1125
114	46	1080i60	1	H	N_0	H	1.000000000	1920	1125
115	47	720p119	1	H	N_0	H	1.000000000	1280	750
116	47	720p120	1	H	N_0	H	1.000000000	1280	750
117	48	480p119	1	T	N_0	T	0.999000999	720	525

118	48	480p119L	1	T	L_{cbb}	H	0.999000999	720	525
119	48	480p120	1	T	N_0	T	1.000000000	720	525
120	48	480p120L	1	T	L_{cbb}	H	1.000000000	720	525
121	49	480p119S	1	T	S	H	0.999000999	720	525
122	49	480p120S	1	T	S	H	1.000000000	720	525
123	50	480i2x59	2	T	N_0	T	0.999000999	720	525
124	50	480i2x60	2	T	N_0	T	1.000000000	720	525
125	50	480i2xL3	2	T	L_{cbb}	H	0.999000999	720	525
126	50	480i2xL4	2	T	L_{cbb}	H	1.000000000	720	525
127	51	480i2xS3	2	T	S	H	0.999000999	720	525
128	51	480i2xS4	2	T	S	H	1.000000000	720	525
129	52	576p200	1	T	N_0	T	1.000000000	720	625
130	52	576p200L	1	T	L_{cbb}	H	1.000000000	720	625
131	53	576p200S	1	T	S	H	1.000000000	720	625
132	54	576i2x_1	2	T	N_0	T	1.000000000	720	625
133	54	576i2xL2	2	T	L_{cbb}	H	1.000000000	720	625
134	55	576i2xS2	2	T	S	H	1.000000000	720	625
135	56	480p239	1	T	N_0	T	0.999000999	720	525
136	56	480p239L	1	T	L_{cbb}	H	0.999000999	720	525
137	56	480p240	1	T	N_0	T	1.000000000	720	525
138	56	480p240L	1	T	L_{cbb}	H	1.000000000	720	525
139	57	480p239S	1	T	S	H	0.999000999	720	525
140	57	480p240S	1	T	S	H	1.000000000	720	525
141	58	480i2x_1	2	T	N_0	T	0.999000999	720	525
142	58	480i2x_2	2	T	N_0	T	1.000000000	720	525
143	58	480i2xL5	2	T	L_{cbb}	H	0.999000999	720	525
144	58	480i2xL6	2	T	L_{cbb}	H	1.000000000	720	525
145	59	480i2xS5	2	T	S	H	0.999000999	720	525
146	59	480i2xS6	2	T	S	H	1.000000000	720	525

Notes:

1. The generator treats double-clocking and pixel repetition as two totally separate items. NCPP controls the number of clocks per pixel, while NPPP controls pixel repetition factor. All library formats set pixel repetition factor NPPP to zero (i.e. OFF) by default. Double-clocking and pixel repetition cannot be applied simultaneously due to AVI:RP field constraints. Therefore, double-clocked formats do not support pixel repetition. Pixel repetition is only applicatlbe to the "4x" formats, where HRES remains at 2880-pixels as the pixel repetition factor NPPP is varied between 1 and 10 – thereby varying the effective resolution. In the case of the formats 480p4x59, 480p4xL1, 480p4xS1, 480p4x60, 480p4xL2, and 480p4xS2, NPPP is only allowed to take on the values 1, 2, or 4.
2. EXCX and EXAR are not listed here, because all library formats set EXCX and EXAR equal to N_0 and CXAR, respectively. These values may be subsequencly changed by rendering a special test image called "AFDtest", after the base format has finished loading, in order to evaluate different AFD cases.
3. TUNE is detuning information. A value of 0.999000999 indicates that the given timing has been tuned by a factor of 1/1.001 for NTSC compatibility. A value of 1.000000000, on the other hand, indicates that the given timing has not been so tuned.
4. Double-clocked formats have the same horizontal resolution as single-clocked formats - the horizontal active (as we define it), is not doubled in the double-clocked case. Some formats are distinguished by a horizontal active that is 4-times the normal value of 720. Here, pixel repetiton may be applied, by a special "PixelRep" test image, after the format has loaded. The "PixelRep" test image allows the number of pixels-per-pixel (NPPP) to be varied and an image with repeated pixels to be rendered for test purposes.
5. Some formats are distinguished by having a slightly different vertical line total.

Digital Video Signal Controls

HDMI High-Definition Multimedia Interface Protocol
0 = disable (use alternate protocol – for example DVI 1.0)
1 = enable (use HDMI 1.0 protocol)

DVPT Digital Video Protocol Type
0 = default (based on hardware present)
1 = DVI 1.0
2 = HDMI 1.0

DVSM Digital Video Sampling Mode
0 = default
1 = 4:1:1
2 = 4:2:2
3 = 4:2:0
4 = 4:4:4

DVST Digital Video Signal Type
0 = none
10 = RGB
13 = YCbCr per SMPTE 260M-1999 Table 1
14 = YCbCr per ITU-R BT.601-5 Table 3, Item 7
15 = YCbCr per ITU-R BT.709-5 Part 1, Section 6.10

DVSS Digital Video Signal Swing
0.0 to 2.0 volts differential peak-to-peak
(nominally 1.000, 0.150 to 1.560 range guaranteed)

GAMC Gamma Correction Enable
0 = no gamma correction
1 = apply gamma correction

GAMA Gamma Correction Factor
2.222… = always

NBPC Number of bits per component
6
8
10
12

DVQM Digital video quantization mode
LMIN? Digital video Minimum Level (unsigned)
LMAX? Digital video Maximum Level (unsigned)

Content Mapping Controls

CXAR is the aspect ratio of the source image content.

SXAR is the aspect ratio normally associated with the current signal format's "coded frame"

EXAR is the extended "shot" aspect ratio.

PXAR is the aspect ratio of pixels in active regions of the raster and matches the values listed in

SXEX Signal-From-Extended Aperture Map
0 to 131071 (see Table 7)

EXCX Extended-From-Content Aperture Map
0 to 131071 (see Table 7)

SXCX Signal-From-Content Aperture Map (sets EXAR=CXAR, SXEX=SXCX, & EXCX=0)
0 to 131071 (see Table 7)

XAFD AFD Mode Setting
0 to 15 sets SXCX, SXEX, EXAR, EXCX, CXAR associated with a given AFD code (see Table 13). This command will not change the value of SXAR, which remains fixed by the current format. Therefore, the range of allowed AFD values are limited to those associated with the value of SXAR in the current format.

Data (Island) Packet Generator (DPG) Parameters

DPGU Data Island Packet Generator Use
Syntax: `DPGU`

DPTG Data Island Packet Type Gate `( gdp=8|audsamples=4|acr=2|gcp=1 )`
Syntax: `DPTG <mask>`

DPTR Data Island Packet Type Repeat Mask `( gdp=8|gcp=1 )`
Syntax: `DPTR <mask>`

(note: audiosamples and acr always repeat when enabled, therefore bit-weights 2 and 4 are a don't care)

MUTE Audio/Video Mute (AVMUTE)
Syntax: `MUTE <state>`
Executing MUTE with <state> = 0 causes "AVMUTE clear flag" to be sent repeatedly, which will cause audio & video to be heard & seen. Executing MUTE with <state> =1 causes "AVMUTE set flag" to be sent repeatedly, which will cause the display to shut OFF audio & video and wait for a format change. Executing a MUTE query will return the current AVMUTE state of the transmitter hardware.

XGCP General Control Parameter Data `(use in combination with the MUTE command/query)`
Syntax: `XGCP <AVMUTE set flag> <AVMUTE clear flag>`

XACR Audio Clock Regeneration Data `(omit CTS value for auto CTS)`
Syntax: `XACR <N value> [<CTS value>]`

XGDP Generic Data Packet Data
SCPI Syntax[5]:

```
XGDP
        [:DATA] <data> |

        [:HB0 <hb0>]

        [:HB1 <hb1>]

        [:HB2 <hb2>]

        [:PB0 <pb0>]

        [:PB1 <pb1>]

        [:PB2 <pb2>]

        .
        .
        .

        [:PB27 <pb27>]
```

where: <hbx> and <pbx> are byte values, <data> is the payload in ASCIIHEX, where a series of ASCII characters '0' thru 'F' are used to represent 4-bit nibbles of data. If less than 60 ASCII characters are given (i.e. < 30 bytes of data), then the remaining nibbles will be automatically set to zero.

InfoFrame Generator (IFG) Parameters

IFGU InfoFrame Generator Use
Syntax: `IFGU`

IFTG InfoFrame Type Gate (gifB=32|mpg=16|aud=8|spd=4|avi=2|gifA=1)
Syntax: `IFTG <mask>`

IFTR InfoFrame Type Repeat Mask (gifB=32|mpg=16|aud=8|spd=4|avi=2|gifA=1)
Syntax: `IFGR <mask>`

XGIF Generic InfoFrame Data
Syntax: `XGIF [ A | B ] <type> <version> <length> <data>`

where: A and B are optional parameters that select first and second generic InfoFrame data buffers, respectively. If A and B are omitted, then the first A buffer is selected by default. Other parameters are type=0 to 127, version=0 to 255, length=0 to 25, , <data> is the payload in ASCIIHEX, where a series of ASCII characters '0' thru 'F' are used to represent 4-bit nibbles of data. If less than 50 ASCII characters are given (i.e. < 30 bytes of data), then the remaining nibbles will be automatically set to zero.

[5] This command/query represents a parameter cluster, whose constituent parts can be individually set and queried using standard SCPI subsystem command syntax. For example, the AVI parameter 'Y' can be set to '1' by using the string "`XAVI:Y 1`". The same parameter can be queried using the string "`XAVI:Y?`". Multiple parameters can be set (in any order) at once using semicolons as follows: "`XAVI:Y 1;M 0;C 1`". Likewise, the same parameters can be queried (in any order) using the string "`XAVI:Y?;M?;C?`". If a header path does not immediately follow a message terminator (e.g. a line-feed), then a leading colon is required to reset the current hierarchical path to the root as in the command strings "`XSPD:VERS 1;:XAVI:Y 1`" and "`XSPD:VERS?;:XAVI:Y?;M?;C?`". See http://www.scpiconsortium.org for details.

XAVI Auxillary Video Information InfoFrame Data
SCPI:

```
XAVI

    [:DATA] <type> <version> <length>
           [<scan info>
            [<bar info>
             [<AFD present>
              [<video type>
               [<active format aspect ratio>
                [<picture aspect ratio>
                 [<colorimetry>
                  [<non-uniform picture scaling>
                   [<end top bar>
                    [<start bottom bar>
                     [<end left bar>
                      [<start right bar>
                       [<video identification code>
                        [<pixel repeat>]
                       ]
                      ]
                     ]
                    ]
                   ]
                  ]
                 ]
                ]
               ]
              ]
             ]
            ]
           ] |

    [:VERS <version>]

    [:S <scan info>]

    [:B <bar info>]

    [:A <AFD present>]

    [:Y <video type>]

    [:R <active format aspect ratio>]

    [:M <picture aspect ratio>]

    [:C <colorimetry>]

    [:SC <non-uniform picture scaling>]

    [:ETB <end top bar>]

    [:SBB <start bottom bar>]

    [:ELB <end left bar>]

    [:SRB <start right bar>]

    [:VIC <video identification code>]

    [:PR <pixel repeat>]

where: type=2, version=1 or 2, and length=13.  Video Identification Code and Pixel
       Repeat parameters are not output unless version=2.

Here are some examples:
```

To output a version 1 AVI InfoFrame with S=1, B=0, A=1, Y=2, R=8, M=1, C=1, SC=0, ETB=0, SBB=0, ELB=0, SRB=0, then send the following:

```
XAVI 2 1 13 1 0 1 2 8 1 1 0 0 0 0 0;IFGU
```

To output a version 2 AVI InfoFrame with S=1, B=0, A=1, Y=2, R=8, M=1, C=1, SC=0, ETB=0, SBB=0, ELB=0, SRB=0, VIC=2, & PR=0, then send the following:

```
XAVI 2 2 13 1 0 1 2 8 1 1 0 0 0 0 0 2 0;IFGU
```

To change the value AVI InfoFrame parameters A and R to 0 (without changing the other parameters of the cluster), then send the following:

```
XAVI:A 0;R 0;IFGU
```

To query the values of AVI InfoFrame parameters B and Y, then send the following:

```
XAVI:B?;Y?
```

DVIC Digital Video Identification Code (see also XAVI:VIC)
0 = unestablished format
1 thru 59 = established format (see Table 18)

XSPD Source Product Description InfoFrame Data
SCPI Syntax (see footnote 5 above):

```
XSPD
    [:DATA] <type> <version> <length>
          [<vendor name string>
           [<product description string>
            [<source device info>]
           ]
          ] |

       [:VERS <version>]

       [:VNS <vendor name string>]

       [:PDS <product description string>]

       [:SDI <source device info>]
```

where: type=3, version=1, and length=25.

XAUD Audio InfoFrame Data
SCPI Syntax (see footnote 5 above):

```
XAUD
        [:DATA] <type> <version> <length>
            [<channel count>
             [1
              [<sample size>
               [<sampling frequency>
                [<channel assignment code>
                 [<level shift value>
                  [<down-mix inhibit>]
                 ]
                ]
               ]
              ]
             ] |
             [<audio coding type>
              [<sample size>
               [<sampling frequency>
                [<maximum bit rate>
                 [<channel assignment code>
                  [<level shift value>
                   [<down-mix inhibit>]
                  ]
                 ]
                ]
               ]
              ]
             ]
            ] |

        [:VERS <version>]

        [:CC <channel count>]

        [:CT <audio coding type>]

        [:SS <sample size>]

        [:SF <sampling frequency>]

        [:MBR <maximum bit rate>]

        [:CA <channel assignment code>]

        [:LSV <level shift value>]

        [:DMI <down-mix inhibit>]

where: type=4, version=1, and length=10.
```

XMPG MPEG InfoFrame Data
SCPI Syntax (see footnote 5 above):

```
XMPG
        [:DATA] <type> version> <length>
            [<MPEG bitrate>
             [<MPEG frame>
              [<field repeat>]
             ]
            ] |

        [:VERS <version>]

        [:MB <MPEG bitrate>]

        [:MF <MPEG frame>]

        [:FR <field repeat>]

where: type=5, version=1, and length=10.
```

Digital Audio Format Parameters

NBPA Number of bits per audio sample (see also `XAUD:SS`)
0 (let incoming external audio stream determine the number of bits per sample)
16
20
24

ARAT Audio Sampling Rate in Hz (see also `XAUD:SF`)
0.0 (let incoming external audio stream determine the audio sampling rate)
192.0E3
176.4E3
96.0E3
88.2E3
48.0E3
44.1E3
32.0E3

BRAT Audio Bit Rate in Hz (see also `XMPG:MB`)
0.0
X.XE3

NDAS Number of digital audio streams
1

NDAC Number of digital audio channels (see also `XAUD:CC`)
2 to 8

DAST Digital Audio Signal Type (see also `XAUD:CT`)
0 = void
1 = IEC 60958-3 Consumer LPCM
2 = IEC 60958-4 Professional LPCM
3 = IEC 61937 w/AC-3 (Dolby Digital)
4 = IEC 61937 w/DTS (Digital Theater Systems)
5 = MP3 (MPEG1 Layer 3)
6 = MP2 (Video CD)
7 = MPEG2 5.1 channel Advanced Audio Coding (AAC)

8 = MPEG2 7.1 channel CBR or VBR
9 = ATRAC

DASI Digital Audio Signal Interface
0 = none
1 = SPDIF
2 = AES3
3 = AES3id
4 = Toslink Optical
5 = MiniPlug Optical

DAXA Digital Audio Content Available
0 to 4095 (see Tables 19 & 20)

DAXG Digital Audio Content Gate
0 to 4095 (see Table 19)

Table 19 - Digital Audio Content Mask

Content	RC7[1]	FRC	FLC	RRC	RLC	RC5[1]	RR	RL	FC[2]	LFE[2]	FR	FL
Bit	11	10	9	8	7	6	5	4	3	2	1	0
Value	2048	1024	512	256	128	64	32	16	8	4	2	1

Table 20 - Digital Audio Content Mapping

Channels												DAXA	DACA	XAUD:CA
7	8	7	8	7	5	6	5	4	3	2	1			
Content														
RC7[1]	FRC	FLC	RRC	RLC	RC5[1]	RR	RL	FC[2]	LFE[2]	FR	FL			
0	0	0	0	0	0	0	0	0	0	1	1	3	3	0
0	0	0	0	0	0	0	0	0	1	1	1	7	7	1
0	0	0	0	0	0	0	0	1	0	1	1	11	11	2
0	0	0	0	0	0	0	0	1	1	1	1	15	15	3
0	0	0	0	0	1	0	0	0	0	1	1	67	19	4
0	0	0	0	0	1	0	0	0	1	1	1	71	23	5
0	0	0	0	0	1	0	0	1	0	1	1	75	27	6
0	0	0	0	0	1	0	0	1	1	1	1	79	31	7
0	0	0	0	0	0	1	1	0	0	1	1	51	51	8
0	0	0	0	0	0	1	1	0	1	1	1	55	55	9
0	0	0	0	0	0	1	1	1	0	1	1	59	59	10
0	0	0	0	0	0	1	1	1	1	1	1	63	63	11
1	0	0	0	0	0	1	1	0	0	1	1	2099	115	12
1	0	0	0	0	0	1	1	0	1	1	1	2103	119	13
1	0	0	0	0	0	1	1	1	0	1	1	2107	123	14
1	0	0	0	0	0	1	1	1	1	1	1	2111	127	15
0	0	0	1	1	0	1	1	0	0	1	1	435	243	16
0	0	0	1	1	0	1	1	0	1	1	1	439	247	17
0	0	0	1	1	0	1	1	1	0	1	1	443	251	18
0	0	0	1	1	0	1	1	1	1	1	1	447	255	19
0	1	1	0	0	0	0	0	0	0	1	1	1539	195	20
0	1	1	0	0	0	0	0	0	1	1	1	1543	199	21
0	1	1	0	0	0	0	0	1	0	1	1	1547	203	22
0	1	1	0	0	0	0	0	1	1	1	1	1551	207	23
0	1	1	0	0	1	0	0	0	0	1	1	1603	211	24
0	1	1	0	0	1	0	0	0	1	1	1	1607	215	25
0	1	1	0	0	1	0	0	1	0	1	1	1611	219	26
0	1	1	0	0	1	0	0	1	1	1	1	1615	223	27
0	1	1	0	0	0	1	1	0	0	1	1	1587	243	28
0	1	1	0	0	0	1	1	0	1	1	1	1591	247	29
0	1	1	0	0	0	1	1	1	0	1	1	1595	251	30
0	1	1	0	0	0	1	1	1	1	1	1	1599	255	31

Notes:

1. Normally, each type of audio content is assigned to a particular channel and is always output on that channel when present. There is one exception to this rule. Rear center (RC) content is switched from channel 5 to 7, whenever rear left (RL) content is simultaneously present. In order to simplify things, RC is therefore treated as if it were two different types of content: RC5 and RC7. RC5 content is simply RC content output on channel 5, whereas RC7 content is the same RC content, but output on channel 7 instead.
2. Note that this ordering is per EIA/CEA-861-C, which unfortunately does not follow established industry standards for channel assignment (e.g. SMPTE 320M-1999 and ITU-R BR.1384).
3. Entering DACA will automatically set DAXA and XAUD:CA parameters to a corresponding value. In the case of the values 243, 247, 251, and 255, the corresponding DAXA and XAUD:CA values are not uniquely determined. In this case, the DAXA and XAUD:CA will be set for the case where XAUD:CA 16, 17, 18, & 19, respectively.
4. If you know what content you want to be available, simply entering DAXA will automatically set both DACA and XAUD:CA to corresponding values. This method should not be used to gate channels. Instead, the DAXG should be used after all of the available channels have been selected.
5. Entering XAUD:CA will automatically set both DAXA and DACA to corresponding values.
6. If either DAXA or DACA are set to a value not found in Table 20, then other parameters will not be automatically set and an error will be generated should the format be loaded while this condition exists.

DACA Digital Audio Channels Available
0 to 255 (see Tables 20 & 21)

DACG Digital Audio Channel Gate
0 to 255 (see Table 21)

Table 21 - Digital Audio Channel Mask

Channel	8	7	6	5	4	3	2	1
Bit	7	6	5	4	3	2	1	0
Value	128	64	32	16	8	4	2	1

DALS Digital Audio Level Shift Value in dBFS (see also `XAUD:LSV`)
0 to 15

DADG Digital Audio Down-mix Gate (see also `XAUD:DMI`)
0=disallow or 1=allow

Sonic Data Generator (SDG) Parameters

SRAT Digital Audio Sinewave Rate in Hz
20.0 to 20.0E3

SCPI Syntax (see footnote 5 above):

```
SRAT
        [:ALL] <frequency of all content> |

        [:DATA] <front left sinewave frequency>
         [<front right sinewave frequency>
          [<low frequency effects sinewave frequency>
           [<front center sinewave frequency>
            [<rear left sinewave frequency>
             [<rear right sinewave frequency>
              [<rear center sinewave frequency>
               [<rear left of center sinewave frequency>
                [<rear right of center sinewave frequency>
                 [<front left of center sinewave frequency>
                  [<front right of center sinewave frequency>]
                 ]
                ]
               ]
              ]
             ]
            ]
           ]
          ]
         ] |

        [:FL <front left sinewave frequency>]

        [:FR <front right sinewave frequency>]

        [:LFE <low frequency effect sinewave frequency>]

        [:FC <front center sinewave frequency>]

        [:RL <rear left sinewave frequency>]

        [:RR <rear right sinewave frequency>]

        [:RC <rear center sinewave frequency>]

        [:RLC <rear left of center sinewave frequency>]

        [:RRC <rear right of center sinewave frequency>]

        [:FLC <front left of center sinewave frequency>]

        [:FRC <front right of center sinewave frequency>]
```

SAMP Digital Audio Sinewave Amplitude in dBFS
-96.3 to 0.0 at NBPA=16-bits
-120.4 to 0.0 at NBPA=20-bits
-144.5 to 0.0 at NBPA=24-bits

SCPI Syntax (see footnote 5 above):

```
SAMP
        [:ALL] <amplitude of all content> |

        [:DATA] <front left sinewave amplitude>
         [<front right sinewave amplitude>
          [<low frequency effects sinewave amplitude>
           [<front center sinewave amplitude>
            [<rear left sinewave amplitude>
             [<rear right sinewave amplitude>
              [<rear center sinewave amplitude>
               [<rear left of center sinewave amplitude>
                [<rear right of center sinewave amplitude>
                 [<front left of center sinewave amplitude>
                  [<front right of center sinewave amplitude>]
                 ]
                ]
               ]
              ]
             ]
            ]
           ]
          ]
         ] |

        [:FL <front left sinewave amplitude>]

        [:FR <front right sinewave amplitude>]

        [:LFE <low frequency effect sinewave amplitude>]

        [:FC <front center sinewave amplitude>]

        [:RL <rear left sinewave amplitude>]

        [:RR <rear right sinewave amplitude>]

        [:RC <rear center sinewave amplitude>]

        [:RLC <rear left of center sinewave amplitude>]

        [:RRC <rear right of center sinewave amplitude>]

        [:FLC <front left of center sinewave amplitude>]

        [:FRC <front right of center sinewave amplitude>]
```

SMAX? Digital Audio Sinewave Maximum Level (unsigned)
=(((2^(NBPA-1))-1)*(10^(SAMP/20)))+(2^(NBPA-1))

SMIN? Digital Audio Sinewave Minimum Level (unsigned)
=((2^(NBPA-1))-1)*(1-(10^(SAMP/20)))

Sonic Data Mixer (SDM) Parameters

SDMG Sonic Data Mixer Gate (external=2|sdg=1)

Syntax: `SDMG <mask>`

Setting the SDMG mask ‘external’ bit to 1 causes the external SPDIF audio signal to be passed through the generator. In this case, the audio InfoFrame CT, CC, SF, and SS parameters are set to zero, indicating that the stream header should be used to determine audio coding type, channel type, sampling frequency, and sample size, respectively. If both ‘external’ and ‘sdg’ bits of the SDMG mask are set to 1 simultaneously, then an error is generated as the generator’s sonic data mixer does not currently support proportional mixing of internal and external sources.

Display Data Channel Power (DCP) Parameters

DCPG Data Channel Power Gate
0=disable +5V power or 1=enable +5V power

DCPX Data Channel Power Overcurrent
0=OK or 1=overcurrent (e.g. shorted)

Appendix A: Automatic InfoFrame Demonstration

A high-level set of format parameters control the settings of the AVI InfoFrame parameters A, R, S, SC, M, B, ETB, SBB, ELB, and SRB in the automatic mode. This set consists of three aperture aspect ratios CXAR, EXAR, SXAR and two aperture-to-aperture maps EXCX and SXEX.

Both CXAR and EXAR represent the shape of program content. Directors must keep either one or two of display screen shapes in mind, when shooting program content. Shooting content for display on a single screen shape is easy and only requires a single content ratio. In this case, both CXAR and EXAR are set equal to the shape of the content. Alternatively, content intended for two different shaped screens could be shot using a technique known as "shoot and protect". Here, CXAR and EXAR are different: CXAR represents "protected" content, that will appear on all displays, while EXAR represents an extended “shot” containing both “protected” as well as extra nonessential imagery for wider or taller displays. Here, the extra nonessential content, which may be masked-off on narrower or shorter displays, must be accounted for separately - hence, the need for two content aspect ratios.

SXAR represents the natural aspect ratio of the video signal format (or coded frame) that transports images to the display.

The two maps EXCX and SXEX fit the three apertures CXAR, EXAR, and SXAR inside one another (see Figure 2). First, the extended-from-content mapping parameter EXCX fits the content aperture CXAR into the extended aperture EXAR. Next, the second signal-from-extended mapping parameter SXEX fits the extended aperture EXAR into the signal aperture SXAR.

In the simplest case, all the apertures match exactly, so the mapping operations essentially "do nothing". First, we’ll look at a situation, where this is the case.

Perform this one-time setup before proceeding with the following examples:

1. Disable all special modes by powering-on with ACS, DCS, DSS keys held down.
2. Connect a dumb terminal and make sure that R:> appears.
3. Make sure you have firmware version equal or greater than 7.48316000(BT) 7.48116000(R), by typing:

```
*IDN?   // should return "QuantumData,802BT-HDMI,8704071,7.48316401"
```

4. Make sure the generator is in the proper power-on modes using SROP command as follows:

```
SROP 10 // put generator in "digital friendly" and "display status" modes
```

5. Make sure that the display (or test fixture) accepts HDMI signal.
6. Make sure that the signal outputs are ON using the OUTPUTS key.

Load the "480p59" format and "Master" image by typing:

```
FMTL 480p59;IMGL Master;ALLU
```

Inspect the high-level content mapping parameters of the format by typing:

```
CXAR? // 1.333...= content aspect ratio is 4:3
EXCX? // 0= extended-from-content is "do nothing"
EXAR? // 1.333...= extended aspect ratio is 4:3
SXEX? // 0= signal-from-extended is "do nothing"
SXAR? // 1.333...= signal aspect ratio is 4:3
```

Note that all of the apertures are set to 4:3 and the maps to zero ("do nothing"). Now, we'll check and see that the automatic AVI InfoFrame parameters are set correctly for this format and image combination by typing the following queries:

```
XAVI:Y?   // 0=RGB
XAVI:A?   // 1=AFD valid
XAVI:B?   // 0=bar data not valid
XAVI:S?   // 2=underscanned (computer)
XAVI:C?   // 1=ITU601 colorimetry
XAVI:M?   // 1=4:3 picture aspect ratio
XAVI:R?   // 8=same as picture aspect ratio
XAVI:SC?  // 0=no known non-uniform scaling
XAVI:ETB? // 0=no horizontal bar present at top of picture
XAVI:SBB? // 481=no horizontal bar present at bottom of picture
XAVI:ELB? // 0=no vertical bar present at left of picture
XAVI:SRB? // 721=no vertical bar present at right of picture
XAVI:VIC? // 2=480p59 or 480p60 EDTV with 4:3 aspect ratio content
XAVI:PR?  // 0=no repetition
```

Here we see the A parameter is set to 1, indicating that the R parameter contains valid AFD data. In this case, R is set to 8, which indicates that the content and signal apertures match (see Table 13 and Table 14, Example 1). The B parameter is set to zero, indicating the bar data is invalid (even though the generator has set it correctly – signaling that no bars are present). The S parameter is set to 2, indicating that the raster is underscanned (i.e. borders are not present). Also, the SC parameter is set to 0, signaling that no known non-uniform scaling has been applied.

Now load the 16:9-in-4:3 letterbox EDTV format "480p59LH" format by typing:

```
FMTL 480p59LH;ALLU
```

Note that the high-level content mapping parameters, of the format, are different now by typing queries:

```
CXAR? // 1.777...= content aspect ratio is 16:9
EXCX? // 0= extended-from-content map not required
EXAR? // 1.777...= extended aspect ratio is 16:9
SXEX? // 264= signal-from-extended map is centered letterbox w/black bars
SXAR? // 1.333...= signal aspect ratio is 4:3
```

Notice that the content of this format is 16:9, while the signal aperture remains at 4:3. This means that a mapping method must be applied to fit the wide16:9 program content into the narrow 4:3 signal aperture. In this case, a letterbox technique (Lcbb) is used to center the 16:9 content within the 4:3 signal aperture, with black bars at the top and bottom. SXEX is set to 264, which sets bits-3 & 8 of the SXEX mask. Bit-3 tells the generator to shrink the 16:9 content to fit the 4:3 signal aperture, while bit-8 puts bars at the top and bottom of the signal aperture (see Table 9 and Table 11). The content is centered, because bits-4 and 5 of SXEX are zero. The bars are black, because bits-6 and 7 are also zero.

Looking at the AVI InfoFrame parameters, once again, we can see that several things have been changed:

```
XAVI:Y?   // 0=RGB
XAVI:A?   // 1=AFD valid
XAVI:B?   // 2=horizontal bar info is valid
XAVI:S?   // 2=underscanned (computer)
XAVI:C?   // 1=ITU601 colorimetry
XAVI:M?   // 1=4:3 picture aspect ratio
XAVI:R?   // 10=16:9 (center)
XAVI:SC?  // 0=no known non-uniform scaling
XAVI:ETB? // 60=last line in top bar is line # 60
XAVI:SBB? // 421=first line in bar at bottom of picture is line # 421
XAVI:ELB? // 0=no vertical bar present at left of picture
XAVI:SRB? // 721=no vertical bar present at right of picture
XAVI:VIC? // 2=480p59 or 480p60 EDTV with 16:9 aspect ratio content
XAVI:PR?  // 0=no repetition
```

Note that the B parameter has been set to 2, indicating that valid horizontal bar info is now present. ETB and SBB bar info parameters now indicate the last and first lines of the top and bottom bars, respectively. Also notice that the value of R has changed to properly indicate the active format being used (see Table 13 and Table 14, Example 7). The value of M remains the same, because the native signal aperture remains 4:3. The VIC code also remains the same as 16:9 content is being shrunk equally in both horizontal & vertical directions to fit the 4:3 signal aperture.

By changing bit-7 to a one (i.e. SXEX to 392), we can change the black bars to white bars as follows:

```
SXEX 392;ALLU // signal-from-extended map is centered letterbox w/white bars
```

Now we'll use the SXEX signal-from-extended high-level mapping format parameter to change the format so the 16:9 extended (content) aperture is mapped into the top of the signal aperture. By changing SXEX, the content can be moved to the top of the raster, which is another standard test case (see Table 13 and Table 14, Example 8):

```
SXEX 280;ALLU // signal-from-extended map is letterbox top w/black bar (Ltbb)
XAVI:Y?   // 0=RGB
XAVI:A?   // 1=AFD valid
XAVI:B?   // 2=horizontal bar info is valid
XAVI:S?   // 2=underscanned (computer)
XAVI:C?   // 1=ITU601 colorimetry
XAVI:M?   // 1=4:3 picture aspect ratio
XAVI:R?   // 2=16:9 (top)
XAVI:SC?  // 0=no known non-uniform scaling
XAVI:ETB? // 0=no horizontal bar present at top of picture
XAVI:SBB? // 361=first line in bar at bottom of picture is line # 361
XAVI:ELB? // 0=no vertical bar present at left of picture
XAVI:SRB? // 721=no vertical bar present at right of picture
XAVI:VIC? // 2=480p59 or 480p60 EDTV with 16:9 aspect ratio content
XAVI:PR?  // 0=no repetition
```

See how the image and InfoFrame parameters again follow suit. The R parameter changes to 2 to indicate a different active format. The ETB changes to 0 to indicate that the top bar is no longer present. SBB is changed to indicate that the bottom bar now starts earlier.

The S parameter can be exercised by introducing a safe-area. Note that this can be done on top of other mapping methods. We can introduce a safe-area by simply changing the value of SXEX again. By setting bits-11 and 15, of the SXEX mask, we can to add a black safe-title border around the entire coded frame (see Table 9 and Table 12, Ktbb).

```
FMTL 480p59;ALLU // reload underscanned library format
SXEX 34816;ALLU  // now apply signal-from-extended map with Ktbb safe-title
XAVI:Y?   // 0=RGB
XAVI:A?   // 1=AFD valid
XAVI:B?   // 3=horizontal and vertical bar info valid
XAVI:S?   // 1=overscanned (for television)
XAVI:C?   // 1=ITU601 colorimetry
XAVI:M?   // 1=4:3 picture aspect ratio
XAVI:R?   // 8=same as picture aspect ratio
XAVI:SC?  // 0=no known non-uniform scaling
XAVI:ETB? // =horizontal bar present at top of picture
XAVI:SBB? // =horizontal bar present at bottom of picture
XAVI:ELB? // =vertical bar present at left of picture
XAVI:SRB? // =vertical bar present at right of picture
XAVI:VIC? // 2=480p59 or 480p60 EDTV with 4:3 aspect ratio content
XAVI:PR?  // 0=no repetition
```

Next, we'll see how the A, R, M and SC parameters change in response to a 16:9 HD-in-SD 4:3 anamorphic squeeze. Load the 480p59SH format and query the AVI parameters once again by typing the following:

```
FMTL 480p59SH;ALLU
```

Inspect the high-level content mapping parameters of the format by typing:

```
CXAR? // 1.777...= content aspect ratio is 16:9
EXCX? // 0= extended-from-content is "do nothing"
EXAR? // 1.777...= extended aspect ratio is 16:9
SXEX? // 1= signal-from-extended via horizontal linear squeeze
SXAR? // 1.333...= signal aspect ratio is 4:3
```

Note that the content apertures are set to 16:9, while the signal remains 4:3. The SXEX map is set to 1 to linearly squeeze the wide16:9 content, along the horizontal axis, in order to fit it into the narrower 4:3 signal aperture (see Table 9 and Table 10). Now, we'll check and see that the automatic AVI InfoFrame parameters are set correctly for this format and image combination by typing queries:

```
XAVI:Y?   // 0=RGB
XAVI:A?   // 0=AFD invalid
XAVI:B?   // 0=bar data not valid
XAVI:S?   // 2=underscanned (computer)
XAVI:C?   // 1=ITU601 colorimetry
XAVI:M?   // 2=16:9 picture aspect ratio
XAVI:R?   // 0=no ADF
XAVI:SC?  // 1=uniform horizontal scaling
XAVI:ETB? // 0=no horizontal bar present at top of picture
XAVI:SBB? // 481=no horizontal bar present at bottom of picture
XAVI:ELB? // 0=no vertical bar present at left of picture
XAVI:SRB? // 721=no vertical bar present at right of picture
XAVI:VIC? // 3=480p59SH or 480p60SH EDTV with 16:9 aspect ratio content
XAVI:PR?  // 0=no repetition
```

The VIC code is changed to 3 and SC parameter is changed to 1 as 16:9 content is being squeezed into the 4:3 signal aperture by non-uniformly scaling in the horizontal direction only.

```
FMTL 1080i29;ALLU
XAVI:Y?   // 0=RGB
XAVI:A?   // 1=AFD valid
XAVI:B?   // 0=bar data not valid
XAVI:S?   // 0=no data
XAVI:C?   // 2=ITU709 colorimetry
XAVI:M?   // 2=16:9 picture aspect ratio
XAVI:R?   // 8=same as picture aspect ratio
XAVI:SC?  // 0=no known non-uniform scaling
XAVI:ETB? // 0=no horizontal bar present at top of picture
XAVI:SBB? // 1081=no horizontal bar present at bottom of picture
XAVI:ELB? // 0=no vertical bar present at left of picture
XAVI:SRB? // 1921=no vertical bar present at right of picture
XAVI:VIC? // 5=1080i29 or 1080i30 HDTV with 16:9 aspect ratio content
XAVI:PR?  // 0=no repetition
```

Conclusion: InfoFrame parameters are automatically calculated by the generator. Therefore, there is (normally) no need to manually program InfoFrame parameters.

Appendix B: Manual InfoFrame Demonstration

show how AVI and AUD parameters can be manually set to anything

```
XAVI:VERS 2
```

Appendix C: AFDtest Image Demonstration

```
// Render the "AFDtest" image by typing:

IMGL AFDtest;ISUB 1;IVER 10;ALLU


XAVI:A?;R?
```

show how B and R parameters are automatically calculated by the generator.

Appendix D: PixelRep Image Demonstration