EBU

OPERATING EUROVISION AND EURORADIO

TECH 3373

COLOUR BARS FOR USE IN THE PRODUCTION OF HYBRID LOG GAMMA (HDR) UHDTV

VERSION 1.0

Geneva
March 2020

Conformance Notation

This document contains both normative text and informative text. All text is normative except for that in the Introduction, any section explicitly labelled as 'Informative' or individual paragraphs which start with 'Note:'.

Normative text describes indispensable or mandatory elements. It contains the conformance keywords 'shall', 'should' or 'may', defined as follows:

'Shall' & 'shall not':	Indicate requirements to be followed strictly and from which no deviation is permitted in order to conform to the document.
'Should' & 'should not':	Indicate that, among several possibilities, one is recommended as particularly suitable, without mentioning or excluding others. OR indicate that a certain course of action is preferred but not necessarily required. OR indicate that (in the negative form) a certain possibility or course of action is deprecated but not prohibited.
'May' & 'need not':	Indicate a course of action permissible within the limits of the document.

Informative text is potentially helpful to the user, but it is not indispensable, and it does not affect the normative text. Informative text does not contain any conformance keywords.

Scope

This document defines a test pattern for use in creating UHDTV video signals. It contains tests for UHDTV, UHDTV to HDTV conversion, measuring luminance response, visualising saturation and hue shifts caused by monitoring equipment, checking correct installation of SMPTE ST 425-5:2019 (Quad 3G-SDI 2 sample Interleaving) equipment and near-black performance. It should NOT be used for setting the brightness control of a reference monitor.

It can be used to:

- Ensure correct settings are applied in hardware
- Highlight errors in transmission chains
- Ensure that transforms from ITU-R BT.2100 HLG to ITU-R BT.709 are correct

Structure of this document

Main document	Describes the Ultra HD Test Signal.
Annex A	Describes the calculation of increasing saturation test patterns for ITU-R BT.2100 primary colours in a hue-invariant colour space.
Annex B	Describes the effect of real hardware interpolation on the accuracy of three dimensional look up tables.
Annex C	Shows the test pattern waveform.
Annex D	Describes the creation of a test pattern for SMPTE ST 425-5:2019 Two Sample Interleave.
Annex E	Describes the command line required to compile the C code into an executable that will create a narrow range 16-bit Y'CbCr image file.

Contents

Colour Bars for use in the production of Hybrid Log Gamma HDR UHDTV

EBU Committee	First Issued	Revised	Re-issued
TC	2020		

Keywords: Test Signal, Professional Video Monitors, Television Production, Colour Bars.

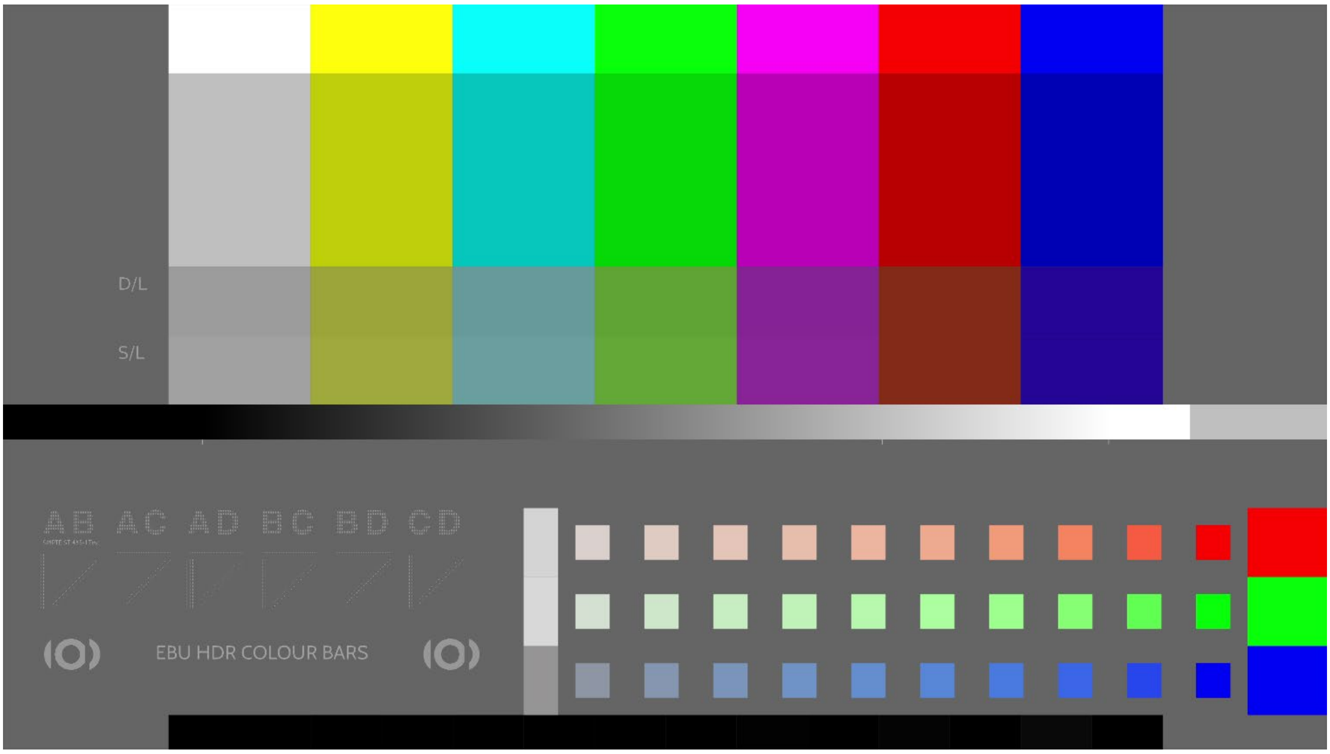


Figure 1: EBU Colour Bars for HLG UHDTV

1. Colour Bars for UHDTV

The colour bar signals defined in this document are designed for measurement of video processing pipelines that originate in the ITU-R BT.2100 HLG [1] format.

Values are given for ITU-R BT.2100 HLG R'G'B' 10-bit narrow-range signals[1]. It is designed to work correctly with Y'CbCr 4:2:2 colour sampling.

The colour bar signals shall not have a channel filter applied prior to use.

Display Light (DL) colour bars should appear correctly on an ITU-R BT.709 [2] vectorscope following a display-light conversion from ITU-R BT.2100 HLG to ITU-R BT.709.

Scene Light (SL) colour bars should appear correctly on an ITU-R BT.709 vectorscope following a scene-light conversion from ITU-R BT.2100 HLG to ITU-R BT.709.

[1] See ITU-R BT.2100-2, table 9.

Two 40% grey areas are provided that can have additional test patterns added, commercial branding colours, for example.

No "2SI Link Test" signal is included in the HD variant.

Figure 2: Textual description of the elements comprising the EBU Colour Bars for HLG UHDTV

Table 1: Pixel dimensions of the elements shown in Figure 2

Dimension	Pixels (UHD)	Pixels (HD)
A	3840	1920
B	2160	1080
C	200	100
D	100	50
E	1160	580
F	480	240
G	412	206
H	206	103
I	476	238

1.1 Colour Bar Signals

These colour bar signals comprise of 100% and 75% ITU-R BT.2100 HLG colour bars and colour bars which will correctly convert to ITU-R BT.709 75% bars when converted using the scene-light and display-light mathematical transforms defined in ITU-R BT.2408 [3]. Values are taken from [4].

Table 3: 100% & 75% HLG colours referenced to ITU-R BT.2100 R'G'B' 10-bit values and the corresponding Y'CbCr values

Colour Name	BT.2100 R' (10-bit)	BT.2100 G' (10-bit)	BT.2100 B' (10-bit)	Y'	Cb	Cr
HDR Colour Bars						
100% White	940	940	940	940	512	512
100% Yellow	940	940	64	888	64	548
100% Cyan	64	940	940	710	637	64
100% Green	64	940	64	658	189	100
100% Magenta	940	64	940	346	835	924
100% Red	940	64	64	294	387	960
100% Blue	64	64	940	116	960	476
75% White	721	721	721	721	512	512
75% Yellow	721	721	64	682	176	539
75% Cyan	64	721	721	548	606	176
75% Green	64	721	64	509	270	203
75% Magenta	721	64	721	276	754	821
75% Red	721	64	64	237	418	848
75% Blue	64	64	721	103	848	485
40% Grey	414	414	414	414	512	512

Table 4: BT.709 (DL converted) colours referenced to ITU-R BT.2100 R'G'B' 10-bit values and the corresponding values Y'CbCr values

Colour Name	BT.2100 R' (10-bit)	BT.2100 G' (10-bit)	BT.2100 B' (10-bit)	Y'	Cb	Cr
BT.709 75% Display Light Colour Bars						
DL White	602	602	602	602	512	512
DL Yellow	594	601	246	578	331	523
DL Cyan	408	591	601	544	543	418
DL Green	388	589	232	515	358	424
DL Magenta	534	227	595	329	656	654
DL Red	522	216	138	292	428	672
DL Blue	187	127	602	171	746	523

Table 5: BT.709 (SL converted) colours referenced to ITU-R BT.2100 R'G'B' 10-bit values and the corresponding Y'CbCr values

Colour Name	BT.2100 R' (10-bit)	BT.2100 G' (10-bit)	BT.2100 B' (10-bit)	Y'	Cb	Cr
BT.709 75% Scene Light Colour Bars						
SL White	618	618	618	618	512	512
SL Yellow	610	616	253	593	327	524
SL Cyan	422	605	615	558	543	418
SL Green	400	603	238	528	354	423
SL Magenta	541	230	601	334	657	656
SL Red	527	218	139	294	427	673
SL Blue	186	126	598	170	745	523

1.2 Luma Ramp

This test signal is designed to measure the effect of processing on the luma (Y') signal. The HLG luma signal is linear. Tone-mapping operators will affect the linearity of the signal and this can be measured on a waveform monitor. Video processing may also limit the range of the signal, causing clipping of highlights and shadows, this can also be measured on a waveform monitor.

The linear ramp covers all values which are legal within a SMPTE SDI signal, i.e. 4-1019. The linear ramp for UHD is 3x1015 pixels wide and for HD is 1.5x1015 pixels wide. The ramp increments linearly and is converted from double precision floating point to integer using a floor function.

The ramp is centred in the test card and the space to the left of the signal is achromatic at code level 64 and to the right is achromatic at code level 721. One-pixel wide tick marks appear at the values 64, 721 and 940.

The luma ramp shall be calculated using this pseudocode:

```
for (int line = rampTop; line <= rampBottom; line++)
  {
    for (int pixel = 0.0; pixel <= aebu; pixel++)
    {
      if (pixel <= blockWidth)
      {
        tchdy[line][pixel] = Plus_0Black_ycbcr[0] * 256;
      }
      else if (pixel > (blockWidth + (3.0 * 1015.0)))
      {
        tchdy[line][pixel] = White_75_ycbcr[0] * 256;
      }
      else if ((pixel > blockWidth)&&(pixel <= (blockWidth + (3.0 * 1015.0))))
      {
        double Yramp = 512.0;
        int pixNum = ((pixel - blockWidth) / 3.0) + 4.0;
        tchdy[line][pixel] = pixNum * 256 / 4.0;
      }
    }
  }
```

1.3 Saturation Test Signals

The saturation test signal is aligned to the Top-Left in the assigned space, the background colour is 40% grey. The following tables give the ITU-R BT.2100 HLG R'G'B' values for the test signal spots. E is replaced by either R, G or B. Appendix A shows how this signal is calculated. This test signal can be used to visually estimate the area of ITU-R BT.2100 gamut that can be displayed by a monitor or television and the effect of video processing on colour reproduction. For convenience, the tables also include the corresponding Y'CbCr values.

Table 6: Red Saturation Test Signals

Colour Name	BT.2100 R' (10-bit)	BT.2100 G' (10-bit)	BT.2100 B' (10-bit)	Y'	Cb	Cr
Red 0.0	793	793	793	793	512	512
Red 0.1	815	779	763	788	499	531
Red 0.2	834	762	730	779	485	550
Red 0.3	851	742	695	768	472	570
Red 0.4	866	718	654	753	458	590
Red 0.5	881	689	607	735	443	614
Red 0.6	894	651	549	709	425	640
Red 0.7	907	600	473	673	403	674
Red 0.8	919	522	386	618	386	721
Red 0.9	930	390	283	526	380	793
Red 1.0	940	64	64	294	387	960

Table 7: Green Saturation Test Signals

Colour Name	BT.2100 R' (10-bit)	BT.2100 G' (10-bit)	BT.2100 B' (10-bit)	Y'	Cb	Cr
Green 0.0	809	809	809	809	512	512
Green 0.1	791	830	783	817	494	494
Green 0.2	770	848	756	822	476	476
Green 0.3	747	863	726	824	459	458
Green 0.4	720	878	691	825	439	439
Green 0.5	688	890	651	823	419	419
Green 0.6	648	902	601	817	394	394
Green 0.7	594	913	536	807	365	364
Green 0.8	511	922	443	786	326	322
Green 0.9	380	931	327	750	282	255
Green 1.0	64	940	64	658	189	100

Table 8: Blue Saturation Test Signals

Colour Name	BT.2100 R' (10-bit)	BT.2100 G' (10-bit)	BT.2100 B' (10-bit)	Y'	Cb	Cr
Blue 0.0	579	578	578	578	512	513
Blue 0.1	552	584	631	578	541	494
Blue 0.2	520	586	677	574	568	475
Blue 0.3	484	584	717	566	594	455
Blue 0.4	445	578	755	554	622	437
Blue 0.5	403	564	790	535	651	420
Blue 0.6	360	542	823	511	682	407
Blue 0.7	312	504	854	474	718	399
Blue 0.8	260	443	884	421	764	400
Blue 0.9	197	347	912	341	822	412
Blue 1.0	64	64	940	116	960	476

1.4 Text Area

The text area is designed to allow the overlay of textual information important for the identification of the signal. Such information could include production name, production location, production date etc.

Fonts should be chosen that are legible and should have an anti-aliasing filter applied. A gap of at least 10 pixels shall be maintained between the edge of the text area and the text.

Further considerations on font choice, e.g. licensing and accessibility requirements, are out of scope of this document.

Table 9: Text Area colours

Colour Name	BT.2100 R' (10-bit)	BT.2100 G' (10-bit)	BT.2100 B' (10-bit)	Y'	Cb	Cr
Background Colour	250	250	250	250	512	512
Text Colour	600	600	600	600	512	512

1.5 Two Sample Interleave (2SI) Test Pattern

Two sample interleave (2SI) is the method outlined in SMPTE ST 425-5:2019 [5] to convey UHD image rasters on four 3G-SDI video cables. Whilst 2SI provides many features, it can be difficult on small production monitors to assess if it is working correctly. This test pattern can identify two major issues: cables being disconnected, and cables being swapped.

1.5.1 Technical Fault 1: Cable disconnected

The 2SI test pattern has several letters displayed (A, B, C, D). If cable A is disconnected, the letter A will not be displayed on the monitor, if cable B is disconnected, the letter B will not be displayed on the monitor, etc.

Figure 3: Technical fault, Cable B is disconnected.

1.5.2 Technical Fault 2: Cables swapped

A further technical fault that can be detected is a cable swap, e.g. cable A being plugged into input B and cable B being plugged into input A. In use this manifests itself as very slight misalignment of pixels on edges and looks like coding noise or aliasing. On a small production monitor (e.g. 30” or less), it can be very difficult to detect, but causes significant issues in encoding and for viewing in the home on larger televisions.

When cables are correctly connected, each horizontal, vertical and diagonal section of the test pattern is comprised of three lines. When a cable swap is present, this is no longer the case, two or four lines are present.

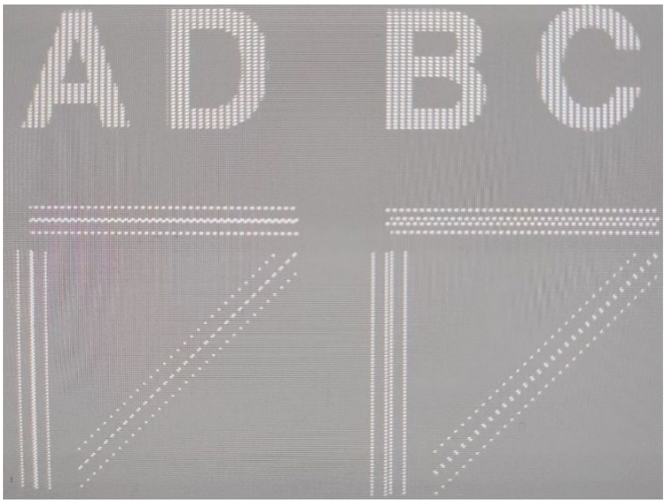


Figure 4: Technical fault, Cable B is connected to input C and Cable C is connected to input B.

Four distinct lines are visible in the horizontal and vertical sections of the BC test pattern section.

1.6 Near-black Test Signal

This test signal shall not be used for setting the black level control (brightness) of a monitor. It shall be used to check that sub-blacks are not removed by equipment in the production chain (e.g. by a legalizer).

Table 10: Near-black Test Signal

Colour Name	BT.2100 R' (10-bit)	BT.2100 G' (10-bit)	BT.2100 B' (10-bit)	Y'	Cb	Cr
-4%	32	32	32	32	512	512
-2%	48	48	48	48	512	512
-1%	56	56	56	56	512	512
0%	64	64	64	64	512	512
1%	72	72	72	72	512	512
2%	80	80	80	80	512	512
4%	96	96	96	96	512	512

2. References

[1] ITU-R BT.2100-2, Image parameter values for high dynamic range television for use in production and international exchange, www.itu.int/rec/R-REC-BT.2100, July 2018.

[2] ITU-R BT.709-6, Parameter values for the HDTV standards for production and international programme exchange, www.itu.int/rec/R-REC-BT.709, June 2015.

[3] ITU-R Report BT.2408-3, Guidance for operational practices in HDR television production, www.itu.int/pub/R-REP-BT.2408, July 2019.

[4] BBC R&D technical document "HLG Look-Up Table Licensing", June 2019.

[5] SMPTE ST 425-5:2019, Image Format and Ancillary Data Mapping for the Quad Link 3 Gbit/s Serial Interface, May 2019.

Annex A: Saturation test signals for EBU HDR Colour Bars.

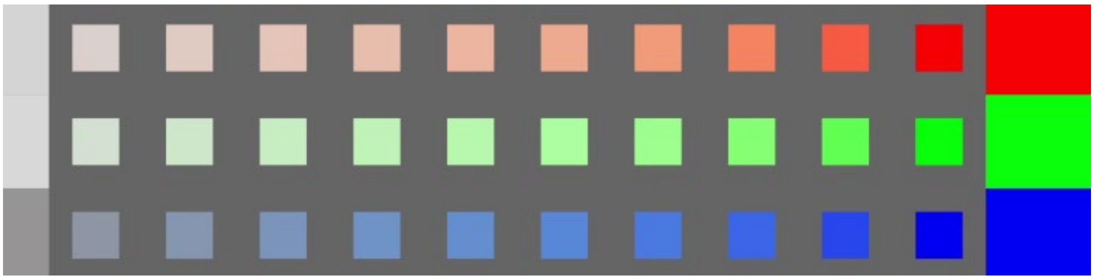


Figure A1: Saturation Test Signal

These test patterns are used for visualising the hue shift inherent in a broadcast monitor or television.

The test patterns show a linearly increasing value for saturation of Red, Green and Blue. 100% Saturation is chosen to be the ITU-R BT.2100 HLG Primary.

Scaling is undertaken in a nominally hue-linear colour space, namely JzAzBz. Jz is constant for each colour, set to the value of Jz used by the 100% primary. Az and Bz Values are calculated in 10% increments using the following code:

```
# IMPORTS
import hlg
import RGB2020
import CIEYuv
import jzazbz

# CONSTANTS
Lw = 1000.0

def tenBitCodeValueRGB(E):
CV = (940.0 - 64.0) * E
    CV = CV + 64.0
    return round(CV)

def printScaled(Jz, Az, Bz):
    for i in range(11):
        a = i / 10.0
        Azn = Az * a
        Bzn = Bz * a
        (X, Y, Z) = jzazbz.calcJzAzBztoXYZ(Jz, Azn, Bzn)
        (CY, Cu, Cv) = CIEYuv.XYZtoCIEYUprimeVprime(X, Y, Z)
        CY = CY * (10000.0 / Lw)
        (X, Y, Z) = CIEYuv.CIEYUprimeVprimetoXYZ(CY, Cu, Cv)
        (R, G, B) = RGB2020.calcXYZtoRGB2020(X, Y, Z)
        R = hlg.HLG_OETF(R)
        G = hlg.HLG_OETF(G)
        B = hlg.HLG_OETF(B)
        R = tenBitCodeValueRGB(R)
        G = tenBitCodeValueRGB(G)
        B = tenBitCodeValueRGB(B)
        print("......Scaled ", a, R, G, B)
```

```
def main( ):
    print("...RED PRIMARY")
    R = 1.0
    G = 0.0
    B = 0.0

    # INVERT HLG OETF
    R = hlg.inverse_HLG_OETF(R)
    G = hlg.inverse_HLG_OETF(G)
    B = hlg.inverse_HLG_OETF(B)

    # Convert to XYZ
    (X, Y, Z) = RGB2020.calcRGB2020toXYZ(R, G, B)

    # Convert to JzAzBz with a peak of Lw cd/m2
    (CY, Cu, Cv) = CIEYuv.XYZtoCIEYUprimeVprime(X, Y, Z)
    CY = CY * (Lw / 10000.0)
    (X, Y, Z) = CIEYuv.CIEYUprimeVprimetoXYZ(CY ,Cu ,Cv)
    (RJz, RAz, RBz) = jzazbz.calcXYZtoJzAzBz(X, Y, Z)

    print("......JzAzBz", RJz, Raz, RBz)

    printScaled(RJz, RAz, RBz)

    print( "...GREEN PRIMARY" )
    R = 0.0
    G = 1.0
    B = 0.0

    # INVERT HLG OETF
    R = hlg.inverse_HLG_OETF(R)
    G = hlg.inverse_HLG_OETF(G)
    B = hlg.inverse_HLG_OETF(B)

    # Convert to XYZ
    (X, Y, Z) = RGB2020.calcRGB2020toXYZ(R, G, B)

    # Convert to JzAzBz with a peak of Lw cd/m2
    (CY, Cu, Cv) = CIEYuv.XYZtoCIEYUprimeVprime(X, Y, Z)
    CY = CY * (Lw / 10000.0)
    (X, Y, Z) = CIEYuv.CIEYUprimeVprimetoXYZ(CY, Cu, Cv)
    (GJz, GAz, GBz) = jzazbz.calcXYZtoJzAzBz(X, Y, Z)

    print("......JzAzBz" , GJz, GAz, GBz)

    printScaled(GJz, GAz, GBz)

    print("...BLUE PRIMARY")
    R = 0.0
    G = 0.0
    B = 1.0
```

```
    # INVERT HLG OETF
    R = hlg.inverse_HLG_OETF(R)
    G = hlg.inverse_HLG_OETF(G)
    B = hlg.inverse_HLG_OETF(B)

    # Convert to XYZ
    (X, Y, Z) = RGB2020.calcRGB2020toXYZ(R, G, B)

    # Convert to JzAzBz with a peak of Lw cd / m2
    (CY, Cu, Cv) = CIEYuv.XYZtoCIEYUprimeVprime(X, Y, Z)
    CY = CY * (Lw / 10000.0)
    (X, Y, Z) = CIEYuv.CIEYUprimeVprimetoXYZ(CY, Cu, Cv)
    (BJz, BAz, BBz) = jzazbz.calcXYZtoJzAzBz(X, Y, Z)

    print("......JzAzBz" , BJz, BAz, BBz)

    printScaled(BJz, BAz, BBz)

if __name__ == '__main__':
    main()
```

Annex B: Effect of LUT hardware on output

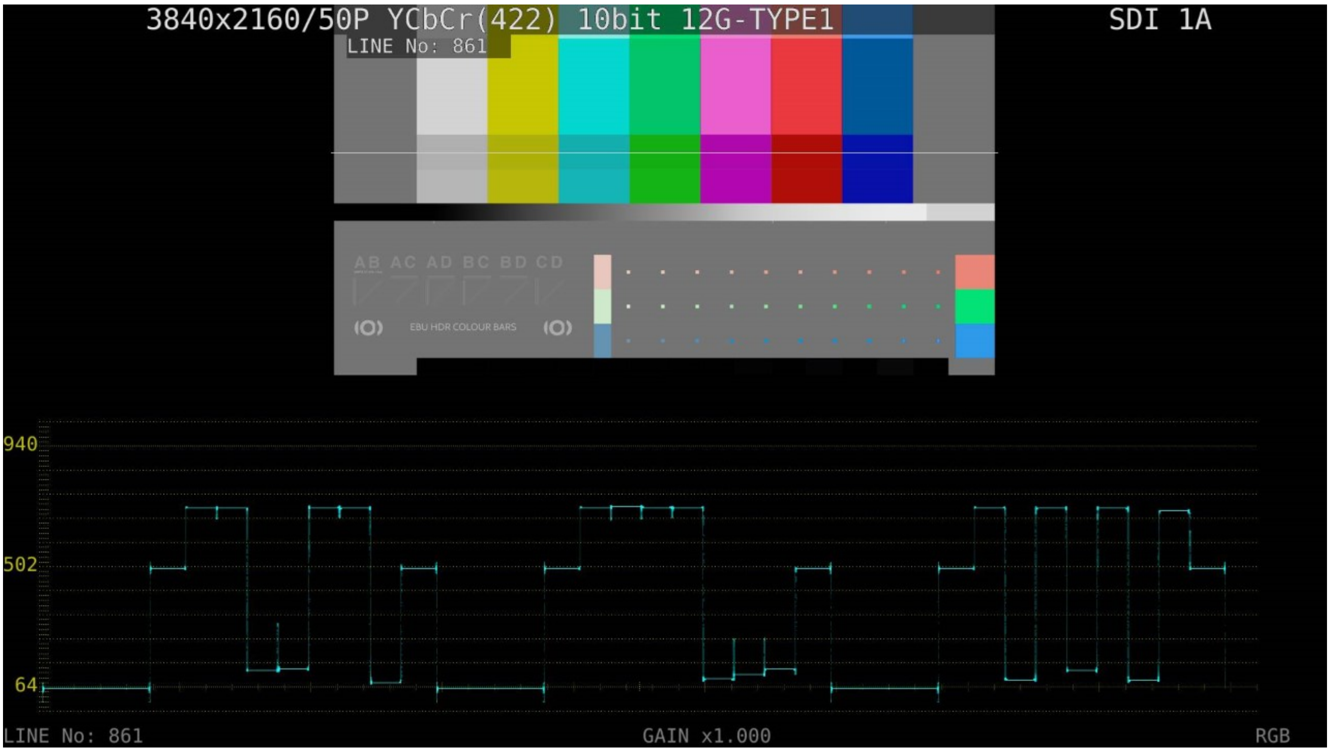


Figure B1: Effect of LUT hardware on output

Three-dimensional Look-Up Tables (LUTs) supply a small number of known points on each colour axis (Red, Green and Blue), typically 17, 33 or 65.

Output values for input triples that do not lie on a point given in the LUT are interpolated.

Inputs to the hardware interpolator must be positive, therefore, near black, interpolation can be incorrect.

For example, if an ideal LUT would provide interpolation points at 1 of -0.05 and 5 of 0.1, a linear interpolator should output a value of 0.07 for an input of 4. If point 1 in the LUT is limited to 0, an output of 0.08 will occur.

Errors of up to 9% have been seen in real hardware implementations.

Annex C: Waveform of test pattern

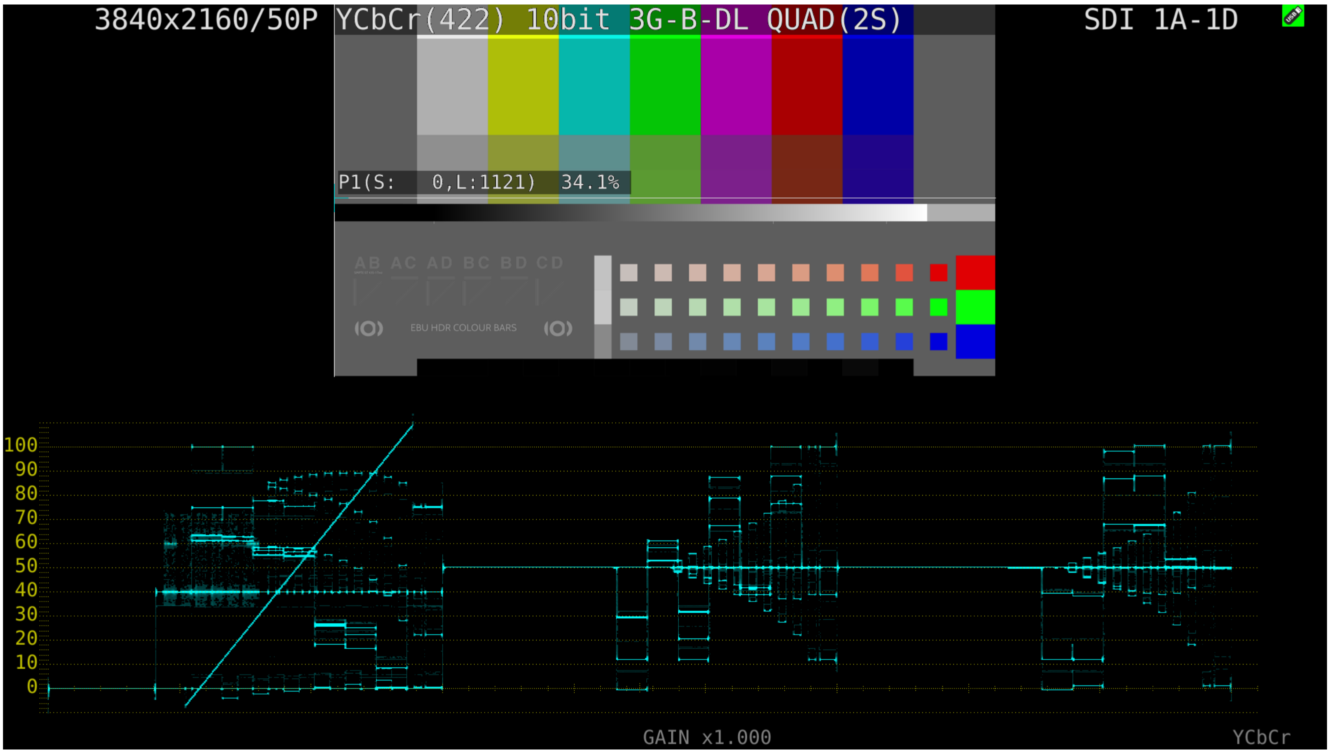


Figure C1: Waveform of test pattern

Annex D: Further information on the Two Sample Interleave test pattern

2-sample interleave (2SI) is a method of conveying a 2160p uncompressed television signal, with a raw data rate of 12 Gbit/s over quad 3 Gbit/s links. It is defined in SMPTE ST 425-5:2019 "*Image Format and Ancillary Data Mapping for the Quad Link 3 Gbit/s Serial Interface*".

The following diagram, taken from the SMPTE document, helps to explain the allocation of luma samples (Y or I) into the sub-images:

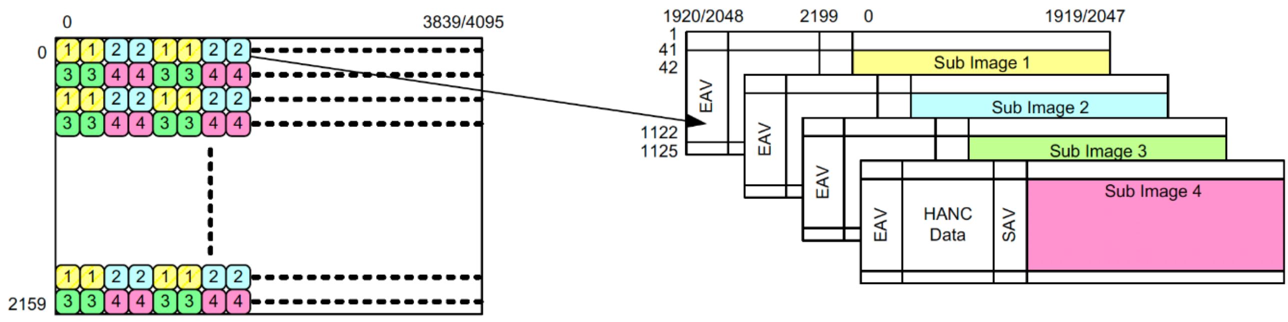


Figure D1: Allocation of luma samples during 2-sample interleave process

The chroma samples (CB, CR or CT, CP) in a 4:2:2 or 4:2:0 system will be allocated on the basis of one of each for every two/four luma samples, and hence any test signal will work best if it only uses luma.

The alternative method is the Quad Split, in which one quarter of the image is simply allocated to each sub-image.

The problem

The problem to be solved is that, if cables are swapped, it can be difficult to spot on a video monitor when using the 2SI method, and it is even harder to diagnose which is swapped. A simpler problem to spot is if one of the four signals is missing, but it is not easily evident which of the four is missing.

The colour bars provide two forms of test signal, one which easily identifies which signal is missing, and one which, if two (or more) are swapped, reveals the problem and identifies which cables are swapped.

Identification of one missing signal

An identification (e.g. a letter "A" or figure "1") is constructed using only the pixels allocated to that sub-image, for each of the sub-images. For the remainder of this document we will use the letters A, B, C & D to represent the sub-images, rather than the numerals 1, 2, 3 & 4.

Identification of swapped cables/signals/images

There are six primary swaps to be considered, A-B, A-C, A-D, B-C, B-D & C-D. Each of these swaps has its own area of the test pattern.

The test for each pair (e.g. A-B swap) is formed of two or three out of:

- Three horizontal lines, one thin (1 pixel wide), one thick (2 pixels wide) and another thin.
- Three vertical lines (again arranged thin, thick, thin).
- Three diagonal lines (again arranged thin, thick, thin).

Each line has “gaps” in it, since it consists of pixels from only one (thin) or two (thick) of the four sub-images. However, on a screen viewed at the standard viewing distance, or even closer, the lines appear essentially continuous.

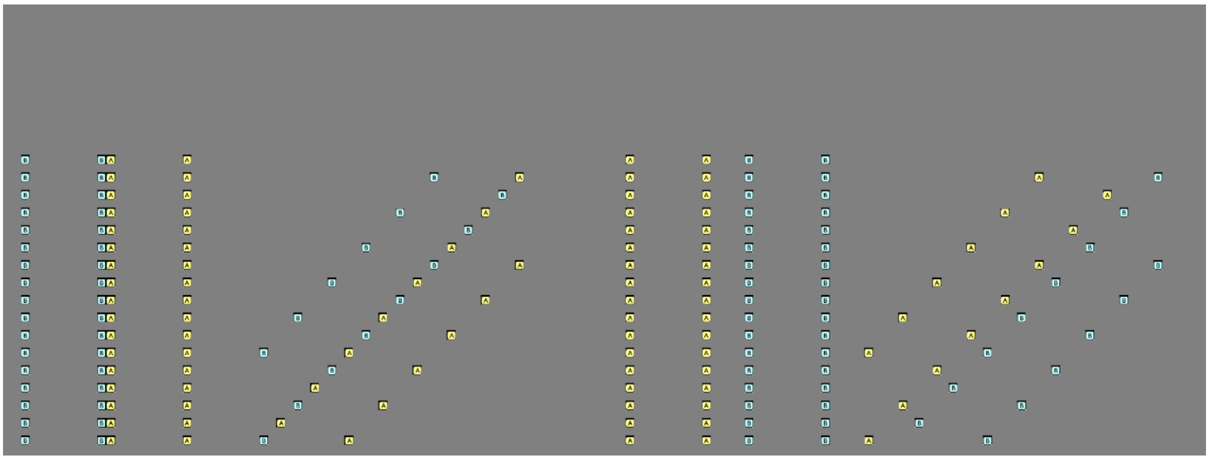
Each of the thin lines only has pixels from one of the two sub-images (e.g. in the A-B pattern, A above or on the left and B below or on the right). These are not strictly necessary for the test but give a visual reference against which the changes to the central wider line are judged.

The central wider line is formed of pixels from adjacent pixels (adjacent vertically, horizontally or diagonally) from the two sub-images in the test area, arranged such that, considering the mapping of groups of 8 pixels in the full image into the pairs of pixels in the sub-images, the pixels allocated to one sub-image fall into a different group of 8 pixels from the pixels allocated to the other sub-image. Thus, if the cables/signals/images being tested are swapped, when the sub-images are re-assembled, the pixels now appear at a distinct separation, giving the appearance of four thin lines rather than the three of the original signal.

If the swap is between a pair where only one of the two in that particular area of the test signal is affected, then the central thicker line, instead of separating to form two distinct thin lines, simply may become slightly blurred.

In the event of a three-way swap, it is usually again possible to see which of the three cables/signals/images are involved since the three of the six test areas where one of the signals is correct will be least affected.

The disposition of pixels in the correct and swapped state may best be understood using an image, as shown below. For some elements of the test signal, the vertical pattern is replaced with a horizontal one.



AB test signal Appearance if A and B are swapped

Figure D2: Identification of swapped cables using a test signal

Annex E: Compiling the colour bar C code

Compile using GCC and the math library:

```
gcc -o EBU_bars EBU_bars.c -lm
chmod a+x EBU_bars
```