Verification Challenges for Deep Color Mode in HDMI

Snigdha Arora, R&D Engineer, Synopsys India Pvt. Ltd, Bangalore, India (snigdhaa@synopsys.com)
Apoorva Mathur, R&D Manager, Synopsys India Pvt. Ltd, Noida, India (apoorvam@synopsys.com)

Abstract— **Color depth, also known as bit depth, is either the number of bits used to indicate the color of a single pixel, in a bitmapped image or video frame buffer, or the number of bits used for each color component of a single pixel. When referring to a pixel the concept can be defined as bits per pixel (bpp). In HDMI (Ref 1, Section 2.3), color depth greater than 24 bits per pixel are defined to be "Deep Color". IP vendors and SoC integrators can have different opinions on the interpretation of this feature, resulting in interoperability issues. It poses both design and verification challenges as well as the need for arriving at a common understanding or interpretation of the requirements for deep color mode. This paper discusses the ambiguities and the correct interpretation based on experience with verifying many IPs and discussions with design and verification engineers. The paper further discusses the verification challenges and key points to recognize and address in order to test and resolve all potential pitfalls.**

Keywords— HDMI, Color depth, Deep Color mode, TMDS.

I. Introduction

For a color depth of 24 bits/pixel, pixels are carried at a rate of one pixel per TMDS clock. At deeper color depths, the TMDS clock is run faster than the source pixel clock providing the extra bandwidth for the additional bits. The TMDS clock rate is increased by the ratio of the pixel size to 24 bits:

- *24 bit mode: TMDS clock= 1.0 x pixel clock (1:1)*
- *30 bit mode: TMDS clock= 1.25 x pixel clock (5:4)*
- *36 bit mode: TMDS clock= 1.5 x pixel clock (3:2)*
- *48 bit mode: TMDS clock= 2.0 x pixel clock (2:1)*

When operating in a Deep Color mode, all video data (pixels) and signaling (HSYNC, VSYNC, DE transitions) are arranged into a series of packed pixel groups, each carrying the same number of pixels and each requiring the same number of TMDS clocks for transmission. On each TMDS clock, one fragment of the pixel group is transmitted. The number of pixels per group and number of fragments per group depends on the pixel size:

- *24 bit mode: 1 pixel/group, 1 fragment/group*
- *30 bit mode: 4 pixels/group, 5 fragments/group*
- *36 bit mode: 2 pixels/group, 3 fragments/group*
- *48 bit mode: 1 pixel/group, 2 fragments/group*

During active video, the input pixel data is packed into these groups. During blanking, HSYNC and VSYNC are packed into these same groups. In this way, all video-related protocol elements are carried at a direct ratio to the pixel clock, thus ensuring no change to the relationship between the pixel clock and the pixel data, DE transitions and HSYNC or VSYNC transitions. This also allows any sequence of HSYNC, VSYNC, DE transitions, that can be supported at 24 bits/pixel to be supported equally in any other pixel size.

As we saw, a pixel group consists of 1, 2, or 4 pixels. Each pixel group is broken into 1, 2, 3 or 5 pixel fragments, and instead of sending each pixel per TMDS clock as in the case of 24 bpp, each fragment is sent per TMDS clock in Deep Color mode.

Each TMDS character period (one TMDS clock) in the transmitted stream carries a single fragment of a pixel group and represents a particular packing phase of the group.
Table 1-4 (Ref 1, Sec 6.5.2) represents the Pixels and Fragments based on different pixel packing phases for all color depths.

24 bit mode: P (pixels/group) = 1 pixel; L (fragments/group) = 1 fragment (1 TMDS character).

Fragment	Phase	Pixels	8 bit HDMI pixel data code (to encoder)							
			Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
8P0	0	A	A0	A1	A2	A3	A4	A5	A6	A7

Table 1: 24 bit mode: P = 1 pixel; L = 1 fragment

Fragment	Phase	Pixels	8 bit HDMI pixel data code (to encoder)							
			Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
10P0	0	A	A0	A1	A2	A3	A4	A5	A6	A7
10P1	1	A+B	A8	A9	B0	B1	B2	B3	B4	B5
10P2	2	B+C	B6	B7	B8	B9	C0	C1	C2	C3
10P3	3	C+D	C4	C5	C6	C7	C8	C9	D0	D1
10P4	4	D	D2	D3	D4	D5	D6	D7	D8	D9

Table 2: 30 bit mode: P = 4 pixels; L = 5 fragments

Fragment	Phase	Pixels	8 bit HDMI pixel data code (to encoder)							
			Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
12P0	0	A	A0	A1	A2	A3	A4	A5	A6	A7
12P1	1	A+B	A8	A9	A10	A11	B0	B1	B2	B3
12P2	2	B	B4	B5	B6	B7	B8	B9	B10	B11

Table 3: 36 bit mode: P = 2 pixels; L = 3 fragments

Fragment	Phase	Pixels	8 bit HDMI pixel data code (to encoder)							
			Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
16P0	0	A	A0	A1	A2	A3	A4	A5	A6	A7
16P1	1	A	A8	A9	A10	A11	A12	A13	A14	A15

Table 4: 48 bit mode: P = 1 pixel; L = 2 fragments

II. Transmission of Pixels for Color Depth Other Than 24bpp

For 30bpp i.e., 10bits per channel, first 8 bits of pixel 1 gets transmitted in fragment 1, remaining 2 bits of pixel 1 are clubbed with 6 bits of pixel 2 and are transmitted in fragment 2, now remaining 4 bits of pixel 2 are clubbed with 4 bits of pixel 3 and are transmitted in fragment 3, remaining 6 bits of pixel 3 are clubbed with 2 bits of pixel 4 and are transmitted in fragment 4, after this the remaining 8 bits of pixel 4 are transmitted in fragment 5 i.e., complete 4 pixels gets transmitted in 5 fragments in case of 30bpp. Same way, for 36bpp, 2 complete pixels gets transmitted in 3 fragments.

Above calculations of pixels and fragments was for the scenario when starting pixel packing phase at the start of video period is "0". Situation becomes more complex when the starting pixel packing phase at the start of Video period is other than "0".

Let's consider the transmission of 4 pixels with starting pixel packing phase as "1" in case of 30bpp.

For PP1, the first 2 bits corresponding to pixel "0" (which is actually not transmitted as complete pixel can never be received, that's why considering it as pixel 0) and 6 bits corresponding to pixel "1" (actually corresponds of 1st pixel) are transmitted in fragment 1, receiver will discard the first 2 bits received in the fragment, as the starting pixel packing phase is phase 1, and the first 8 bits of the pixel "0" are not received and hence the 2 bits received in 1st fragment are of no use. After this, the remaining 4 bits of pixel "1" along with 4 bits of pixel "2" are transmitted in fragment 2, and then remaining 6 bits of pixel "2" are clubbed with 2 bits of pixel "3" and are transmitted in fragment 3, further remaining 8 bits of pixel "3" transmitted in fragment 4, then for transmission of 4th pixel, first 8 bits gets transmitted in fragment 5, still 2 bits of pixel "4" are left, which further requires transmission of one extra fragment, thus the remaining 2 bits of pixel "4" gets transmitted in 6th fragment in this case.

Same way for 36bpp i.e., 12bits per channel, we require 4 fragments to transmit 2 complete pixels in case when pixel packing phase at start of video period is "1", whereas in normal scenario 2 complete pixels gets transmitted in 3 fragments.
Let's see how we can compensate this increase in video fragment.

III. PROBLEM STATEMENT

To identify a way to compensate the increase in video period fragments with a decrease in fragments of blanking period keeping the total number of pixels intact, in case starting pixel packing phase at start of video period is other than PP0. We need to make sure that even after decreasing the fragment from a particular component of horizontal blanking we will still be able to transmit the complete information i.e., need to do verification regarding the complete transmission of component which is used for compensating the extra fragment in video data.

IV. SOLUTION

Let's take an example of ***VIC=2***, having following pixel values for different horizontal regions of a line.
Hactive=720 Hfront=16 Hsync=62 Hback=60 Htotal=858

A. 48 bit mode

In case of 48 bpp, possible pixel packing phase at start of Video period is 0 only as if pixel packing phase at start of Video period is 1, then we will never able to get that complete pixel. So, PP1 at start of video period is not possible in case of 48 bpp. That's why there is no challenge of compensating the pixel and verifying the same in case of 48bpp.

B. 36 bit mode

In case of 36 bpp, possible pixel packing phase at start of Video period are 0 and 1.
Numbers of fragments in case of 36 bit color depth are equal to 1.5 times the number of pixels.

Calculation for horizontal fragments:

If pixel packing phase = 0 at the start of video data then the value of fragments are as follows:
*Hactive = 720 * 1.5 = 1080*
*Hsync = 62 * 1.5 = 93*
*Hback =60 * 1.5 = 90*
*Hfront = 16 * 1.5 = 24*
*Htotal = 858 * 1.5 = 1287*
Htotal_1 = (Hactive+ Hback +Hsync+ Hfront) = 1080+183+24 = 1287
Htotal == Htotal_1

If pixel packing phase = 1 at the start of video data then the value of fragments are as follows:

We need one extra fragment in video data as the first fragment of video does not contain the valid data in first 4 bits and needs to be discarded. Therefore an extra fragment is required for complete transmission of video pixels.

*Hactive = 720 * 1.5 = 1080 + 1 (Extra fragment to compensate loss of first 4 bits) = 1081*
*Hsync = 62 * 1.5 = 93*
*Hback =60 * 1.5 = 90*
*Hfront = 16 * 1.5 = 24*
*Htotal_1 = 858 * 1.5 = 1287*
Htotal = (Hactive+ Hback +Hsync+ Hfront) = 1081+183+24 = 1288
Htotal != Htotal_1
Since Htotal != Htotal_1, Htotal i.e., width of frames should remain same irrespective of change in pixel packing phase.

To resolve this problem we need to adjust the extra fragment required in Hactive by decrease in Hfront's fragment. Hence the calculations for pixel packing phase 1 are modified as:

*Hactive = 720 * 1.5 = 1080 + 1 = 1081*
*Hback+Hsync (Blank pixels) = 122(62+60) * 1.5 = 183*

*Hfront = 16 * 1.5 = 24 – 1 = 23 (To compensate the increase in fragment for Hactive)*
*Htotal_1 = 858 * 1.5 = 1287*
Htotal = (Hactive+ Hback+Hsync+ Hfront) = 1081+183+23 = 1287
Now Htotal == Htotal_1

Let's see how by decreasing 1 fragment in Hfront, we will still be able to transmit complete Hfront pixels. The fragments states during control period in case of 36 bit are shown in Table 5 (Ref 1, Sec 6.5.2):

Fragment	HS/VS value
12C0	S
12C1	T
12C2	T

Table 5: Group size = 2 pixels; 3 fragments

In this case, when pixel packing phase at start of video period is "1", total 1081 Hactive fragments are transmitted, after transmitting total number of 1081 fragments, the pixel packing phase at the start of control period i.e., at start of Hfront is "2". Thus, Hfront starts from 12C2 state, transmitting 1st complete pixel in 1st fragment, followed by 2 pixels getting transmitted in 3 fragments till 15th pixel, and the last pixel i.e., 16th pixel gets transmitted in 12C0 state as indicated in Figure 1 :

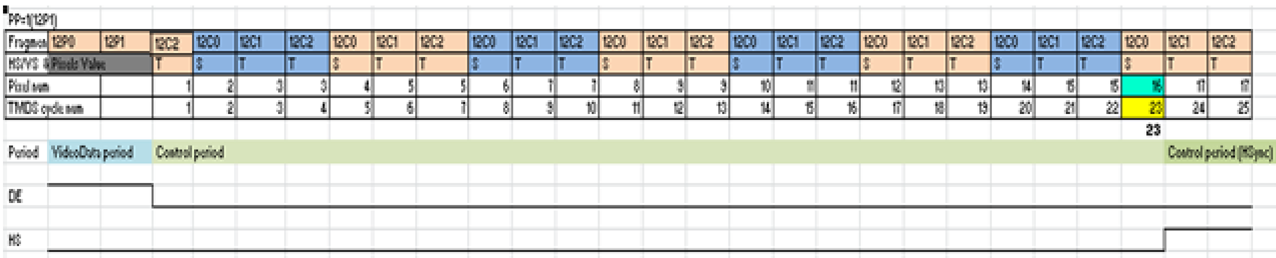

Figure 1: Fragments and Pixels of Hfront in case of PP1 for 36 bit Color depth

Calculating the total Hfront fragments this way are: 1 (1st pixel) + 14 * 1.5 (next 14 pixels) + 1 (Last 16th pixel) = 1+21+1 = 23 fragments.

As we saw, the increase in Hactive fragment gets compensated by decrease in Hfront fragment and still we are able to transmit the complete Hfront information.

Complete functional coverage is a key metric used to ensure verification closure. For ensuring complete verification of all the cases discussed in the sections above, coverage model needs careful thought. In addition to covering the bits involved in sending video frames supporting deep color mode may not be sufficient for complete confidence. A better and comprehensive approach is to define scenario coverage. The focus of testing would then be to achieve 100% scenario coverage.

By default, the transmission of video data starts with pixel packing phase as "0". This is the ideal scenario in most cases as it does not require any compensation in video data. The complete video data gets transmitted in hactive * 1.5 fragments requiring no extra fragment for compensation. Sequences need to be created where video data start with pixel packing phase as "1" leading to the requirement of extra fragment in transmission exercising the scenario requiring the compensation by horizontal blanking period.

C. 30 bit mode

VIC=2 (Hactive= 720   Hfront=16   Hsync=62   Hback=60   Htotal=858)

In case of 30 bpp, possible pixel packing phase at start of video period are 0, 1, 2 & 3.
Numbers of fragments in case of 30 bit color depth are equal to 1.25 times the number of pixels as indicated in Table 2.

Calculation for horizontal fragments:
*Hactive = 720 * 1.25 = 900*
*Hback= 60 * 1.25 = 75*

*Hsync = 62 * 1.25 = 77.5*
*Hfront = 16 * 1.25 = 20*
*Htotal = 858 * 1.25 = 1072.5*

As we can see, Hsync and Htotal components are fractional; therefore in this case calculation of fragments becomes a bit more complex.
Let's look at each component one by one.

Hactive: Number of fragments for Hactive is 900 if starting pixel packing phase at start of Video period is 0. If the starting pixel packing phase at start of video period is 1, 2 or 3 i.e., other than 0, then an extra fragment is required for compensating the loss of 2 bits if the video period starting pixel packing phase is 1, loss of 4 bits if pixel packing phase is 2, or loss of 6 bits if pixel packing phase is 3 at start of the Video period. Number of Hactive fragments in this scenario is 900+1 = 901.

Hback: Number of fragments for Hback is 60 * 1.25 = 75 in all scenarios.

Hsync: Number of fragments for Hsync is 62 * 1.25 = 77.5, i.e.; a fractional number, and there is no way we can transmit a fractional value on TMDS lines, in this scenario, we will transmit 77 and 78 fragments respectively on alternate lines.

Hfront: Number of fragments for Hfront is 16 * 1.25 = 20, when pixel packing phase at start of video period is 0. And as we saw in case of color depth 36 bit, the extra fragment required in Hactive should be compensated with decrement of a fragment from Hfront, same way for 30bpp the increase in Hactive fragment is compensated by the decrease in Hfront fragment when pixel packing phase at the start of video period is 1, 2 or 3. Thus the number of Hfront fragments is 16*1.25 = 20 - 1 = 19. In next section we will see, how complete Hfront is getting transmitted in 19 fragments.

Htotal: Number of fragments for Htotal is 858 * 1.25 = 1072.5 i.e.; a fractional number, and there is no way we can transmit a fractional value on TMDS lines therefore 1072 and 1073 fragments will be transmitted on alternate lines in this case. And as we know Htotal is total of Hsync+Hback+Hactive+Hfront. If we total fragments of Hsync, Hback, Hactive & Hfront, then the Htotal will be 1072 and 1073 on alternate lines.

If PP0
Htotal = (Hactive+ Hback +Hsync+ Hfront) = 900+77+20 = 1072
Htotal = (Hactive+ Hback +Hsync+ Hfront) = 900+78+20 = 1073

If PP1, PP2 or PP3
Htotal = (Hactive+ Hback +Hsync+ Hfront) = 901+77+19 = 1072
Htotal = (Hactive+ Hback +Hsync+ Hfront) = 901+78+19 = 1073

Transmission of Hfront

Let's see how by decreasing 1 fragment in Hfront, we will still be able to transmit complete Hfront pixels.

The fragments states during control period in case of 30 bpp are shown in Table 6 (Ref 1, Sec 6.5.2):

Fragment	HS/VS value
10C0	S
10C1	T
10C2	U
10C3	V
10C4	V

Table 6: Group size = 4 pixels; 5 fragments

States defined in Table 6 are followed in normal scenario, and when pixel packing phase at the start of Hfront is 0 or 1. But when falling edge of DE occurs in mid group i.e., when pixel packing phase at start of Hfront (end of video period) is 2, 3 or 4, the states followed are shown in Table 7 (Ref 1, Sec 6.5.2):

Fragment	HS/VS value
10PC2	T
10PC3	U
10PC4	V

Table 7: 30-bit mode remnant (Falling Edge of DE occurs mid-group). Bridge states for transition from 10Pn to 10C0 are named “10PCn”

Transmission of Hfront fragments:

1) Pixel packing phase at start of video period is “0”.

In this case, when pixel packing phase at start of video period is “0”, Hactive fragments are 900. After transmitting total number of 900 fragments, the pixel packing phase at start of control period i.e.; at start of Hfront is “0”. Thus Hfront starts from 10C0 state, transmitting complete 16 pixels in 20 fragments as indicated in Figure 2.

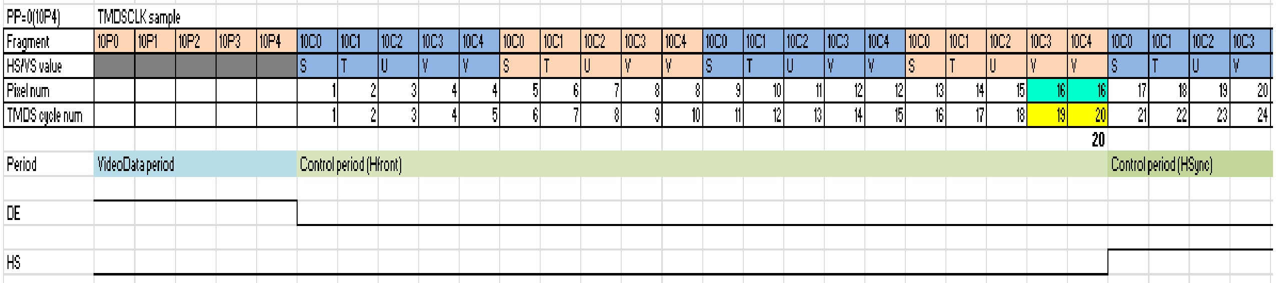

Figure 2: Fragments and Pixels of Hfront in case of PP0 for 30 bit color depth

As we saw, 16 pixels of Hfront are getting transmitted in 20 fragments, i.e., 4 pixels are getting transmitted in 5 fragments.

Validation in this case is quite simple as by default the transmission of video data start with pixel packing phase as “0”. This is the ideal scenario as it does not require any compensation in video data. The complete video data gets transmitted in hactive * 1.25 fragments requiring no extra fragment for compensation.

2) Pixel packing phase at start of video period is “1”.

In this case, when Pixel packing phase at start of Video is “1”, Hactive fragments are 901.After transmitting total number of 901 fragments, the pixel packing phase for start at control period i.e.; at start of Hfront is “2”. Thus Hfront starts from 10PC2 state, transmitting 1, 1 & 1 complete pixel in 10PC2, 10PC3 and 10PC4 states respectively, leading to transmission of first 3 pixels in 3 fragments. This is followed by 4 pixels getting transmitted in 5 fragments each till 15th pixel, and the last pixel i.e., 16th pixel gets transmitted in 10C0 state as indicated in Figure 3 :

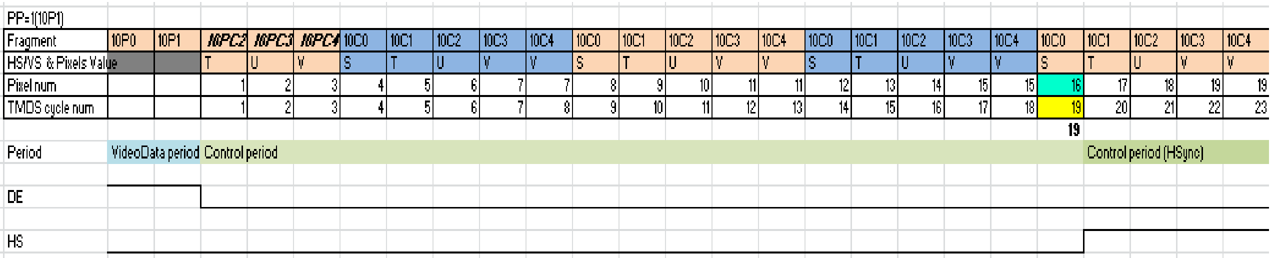

Figure 3: Fragments and Pixels of Hfront in case of PP1 for 30 bit color depth

Calculating the total Hfront fragments this way are: 3 (first 3 pixels) + 12 * 1.25 (next 12 pixels) + 1 (Last 16th pixel) = 3+15+1 = 19 fragments.

To validate above case, we should have scenarios where video data start with pixel packing phase as “1” leading to the requirement of extra fragment in transmission exercising the scenario requiring the compensation by horizontal blanking period.

3) Pixel packing phase at start of video period is “2”.

In this case, when Pixel packing phase at start of Video is “2”, Hactive fragments are 901.After transmitting total number of 901 fragments, the pixel packing phase for start at control period i.e.; at start of Hfront is “3”.

Thus Hfront starts from 10PC3 state, transmitting 1, 1 complete pixel in 10PC3 and 10PC4 states respectively, leading to transmission of first 2 pixels in 2 fragments. This is followed by 4 pixels getting transmitted in 5 fragments each till 14th pixel, and the last two pixel i.e., 15th & 16th pixel gets transmitted in 10C0, 10C1 states respectively, as indicated in Figure 4 :

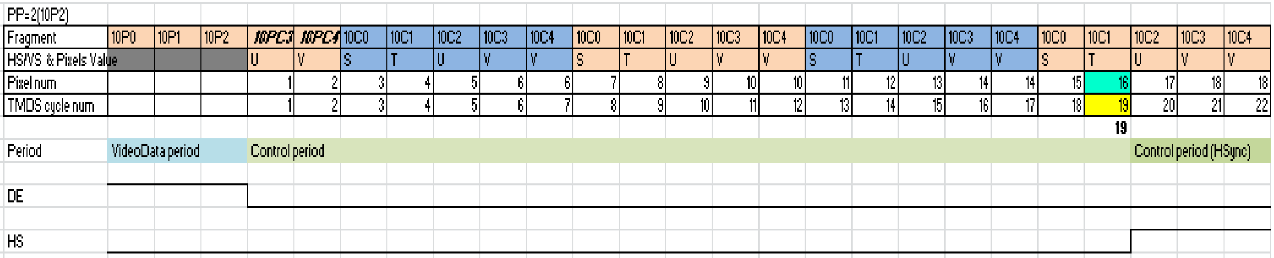

Figure 4: Fragments and Pixels of Hfront in case of PP2 for 30 bit Color depth

Calculating the total Hfront fragments this way are: 2 (first 2 pixels) + 12 * 1.25 (next 12 pixels) + 2 (Last 2 pixels) = 2+15+2 = 19 fragments.

To validate above case, we should have scenarios where video data start with pixel packing phase as “2” leading to the requirement of extra fragment in transmission exercising the scenario requiring the compensation by horizontal blanking period.

4) Pixel packing phase at start of Video is “3”.

In this case, when Pixel packing phase at start of Video is “3”, Hactive fragments are 901.After transmitting total number of 901 fragments, the pixel packing phase for start at control period i.e.; at start of Hfront is “4”. Thus Hfront starts from 10PC4 state, transmitting 1 complete pixel in 10PC4 state followed by 4 pixels getting transmitted in 5 fragments each till 13th pixel, and the last three pixel i.e., 14th, 15th & 16th pixels gets transmitted in 10C0, 10C1 & 10C2 states respectively, as indicated in Figure 5 :

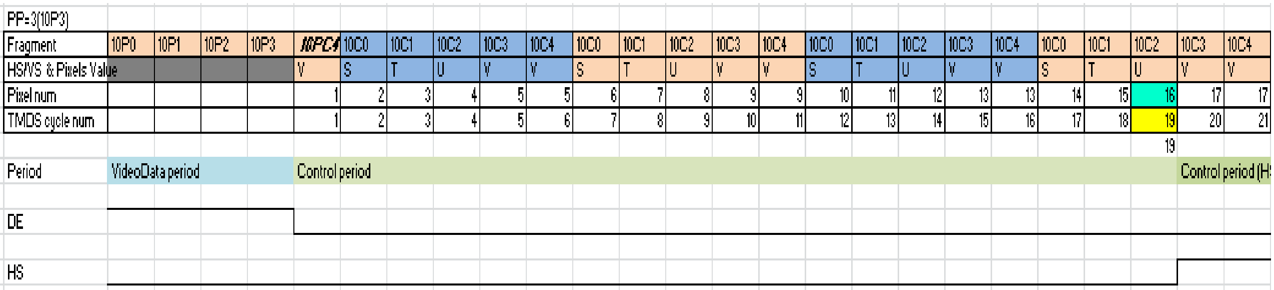

Figure 5: Fragments and Pixels of Hfront in case of PP3 for 30 bit Color depth

Calculating the total Hfront fragments this way are: 1 (first pixel) + 12 * 1.25 (next 12 pixels) + 3 (Last 3 pixels) = 1+15+3 = 19 fragments.

To validate above case, we should have scenarios where video data start with pixel packing phase as “3” leading to the requirement of extra fragment in transmission exercising the scenario requiring the compensation by horizontal blanking period.

V. PROTOCOL CHECKS AND COVERAGE

A. Protocol Checks

There should be appropriate and enough assertions or temporal/procedural checks for all the combinations of different color depths and different pixel packing phases.

Checks are required for video data fragments for all pixel packing phases supported by that color depth and to ensure the increase in fragment required for complete transmission of video active period is getting compensated by correct component of Horizontal blanking period.

There should be checks related to Hsync rising and falling edge, i.e., for the period of Hsync to ensure that the extra fragment is not getting compensated by Hsync fragments.

There should be checks for ensure that the pixel packing phase is getting adjusted properly when there is a difference in the pixel packing phase of transmitter and receiver.

B. Coverage

As discussed above doing Signal level coverage, data object and transection class variables coverage is not sufficient for verification closure. There is a need for comprehensive scenario coverage for as well as cross coverage for pixel packing phase, color mode, stating phase and other contributing attributes.
Here's the sample of coverage code:

```
covergroup coverage_for_deep_color_mode @(event_for_deep_color_mode);
  type_option.comment = "Coverage of GCP Packet";
  option.per_instance = 1;

  color_depth : coverpoint color_depth {
    type_option.comment = "Coverage of Color Depth";
    option.weight = 0;
    bins color_depth_24_bit = {4'b0100};
    bins color_depth_30_bit = {4'b0101};
    bins color_depth_36_bit = {4'b0110};
    bins color_depth_48_bit = {4'b0111};
  }

  pxl_packing_phase : coverpoint pxl_packing_phase{
    type_option.comment = "Coverage of Pixel Packing Phase";
    option.weight = 0;
    bins pxl_packing_phase_1 = {1};
    bins pxl_packing_phase_2 = {2};
    bins pxl_packing_phase_3 = {3};
    bins pxl_packing_phase_4 = {0};
  }

  color_depth_pxl_packing_phase_cross: cross color_depth,
                                               pxl_packing_phase
  iff(rcvd_gcp_pkt == 1'b1) {
    type_option.comment = "Cross Coverage between Color Depth (CD Field) and Pixel Packing phase";
    option.weight = 1;
    ignore_bins cross_cd_24 = binsof(pxl_packing_phase) && binsof(color_depth.color_depth_24_bit);
    ignore_bins cross_cd_36_pp_3 =  binsof(color_depth.color_depth_36_bit) && binsof(pxl_packing_phase.pxl_packing_phase_3);
    ignore_bins cross_cd_36_pp_4 =  binsof(color_depth.color_depth_36_bit) && binsof(pxl_packing_phase.pxl_packing_phase_4);
    ignore_bins cross_cd_48_pp_2 =  binsof(color_depth.color_depth_48_bit) && binsof(pxl_packing_phase.pxl_packing_phase_2);
    ignore_bins cross_cd_48_pp_3 =  binsof(color_depth.color_depth_48_bit) && binsof(pxl_packing_phase.pxl_packing_phase_3);
    ignore_bins cross_cd_48_pp_4 =  binsof(color_depth.color_depth_48_bit) && binsof(pxl_packing_phase.pxl_packing_phase_4);
  }
endgroup
```

VI. CONCLUSION

For deep color mode, when pixel packing phase at start of video period is other than 0, then we require an extra fragment to transmit complete video data This increase in fragment count for video data is getting compensated by decrease in Hfront fragment and still able to transmit complete Hfront information. Assertions or protocols checks and functional coverage should be written in such a way that it exercises and checks all the complex scenario's required for verification of the implemented logic.

REFERENCES

[1] High-Definition Multimedia Interface Specification Version 1.4b

[2] A DTV Profile for Uncompressed High Speed Digital Interfaces ANSI/CEA-861-F