VESA STANDARDS CHANGE REQUEST FORM

To be Filled in by Submitter (Refer to VESA Document VP235I, Section 5)

TITLE:	DSC 1.2b C model improvements and narrow slice warning SCR
AFFECTED DOCUMENT:	DSC 1.2b
REVISION CATEGORY:	Category 1
SUBMITTED TO:	Task Group
SPONSOR:	Fred Walls, Broadcom

SCR REVISION HISTORY	
(DATE)	**(CHANGE)**
11/16/2021	**Initial Submission of SCR**
11/22/2021 (draft 2)	Updated C model with better warning message; updated SCR with better background information.
12/15/2021 (r3)	Updated C model to remove debug message and fix compiler warnings. Reworded paragraph to remove "rule of thumb"

(add more rows as needed)

To be Filled in by VESA Office:

VESA SCR NUMBER:	(To be assigned by VESA office)
SCR ENTRY DATE:	11/16/2021

To be Filled in by Task Group or VESA Office

SCR ADOPTED, REJECTED, or otherwise DISPOSITIONED for other action	SCR is (adopted) or (rejected) or (Dispositioned for other action) If rejected, explain reason for acceptation or rejection If dispositioned, explain action or plan for action (such as including in future draft specification revision, or re-visiting at future date, or other)
DATE SCR ADOPTED or REJECTED or DISPOSITIONED	ADOPTED 3/4/2022

Summary of the Proposed Change(s)

A few C model improvements have been identified since the adoption of DSC 1.2b:

- When reading a .DSC file specified in the input list of files, do not prepend OUT_DIR. This allows the use of fully qualified path names for .DSC files in the input list.
- Modified command parsing to allow for spaces (e.g., spaces in directory names or file names, or spaces in parameter lists).
- When FUNCTION = 2, look for a .ref.ppm (or .dpx or .yuv, depending on the config file settings) in the output directory for PSNR computation. This allows PSNR computation when running FUNCTION = 1 followed by FUNCTION = 2.
- Fixed a bug where some PPS config file parameters previously needed to be correctly specified when FUNCTION = 2 even though the PPS was included in the DSC file.
- Corrected some PPS parameter names when printing the PPS ("ofs" => "offset")
- Fixed a bug in the yuv_read() function where each Y sample was read twice (duplicated code).
- Improved max_QP[14] checker to work with narrow slice widths (considers the impact of unallocated bits at the ends of slice lines). Added GENERATE_RC_PARAMETERS = 2 option to allow users an easy way to get around the issue.

IPR (Intellectual Property Rights) declaration, if any

N/A

Benefits as a Result of the Changes

These changes improve the usability of the C model in certain scenarios and fix some minor bugs.

Assessment of the Impact

These changes are not normative and are intended to improve usability without affecting legacy behavior.

Analysis of the Device Hardware Implication

No impact.

Analysis of the Device Software Implications

No impact.

Analysis of the Compliance Test & Interop Implications

No impact.

New Referenced Documents Resulting from Change

DSC spec updated to reference new C model (DSC_model_20211122.zip).

Attachments

See: https://groups.vesa.org/wg/DSC/document/17478

Proposed Document Change(s) or Addition(s)

Edit Table 1-7 as follows:

…

VESA DSC C Model[b]	Version ~~1.63~~1.67	~~June 23~~December 13, 2021	*DSC C Model*

…

Background Information

Peraton Labs reported a usability issue with the DSC model that the SRC_LIST file could not be configured to read in .DSC files with fully qualified path names, because the model assumes that DSC files are always located in the OUT_DIR directory. Peraton also reported that spaces were not allowed in configuration file parameters. Therefore, this SCR updates the model to allow for fully-qualified path name in .DSC files and for spaces in configuration file parameters.

NVidia reported RCB overflow errors occurring with specific configurations where a narrow slice width was used and RC parameters were automatically generated (i.e., GENERATE_RC_PARAMETERS set to 1). Upon further investigation, it was found that the existing ***rc_range_parameters[14].range_max_qp*** warning was not taking into account additional bits required over the course of a line to make up for unbudgeted bits at the end of a line. In DSC, most groups consist of 3 pixels (or pixel pairs), but the last group of a line might consist of 1, 2, or 3 pixels, depending on the slice width. When the last group consists of either 1 or 2 pixels, the number of bits budgeted for that group is either ***bits_per_pixel*** or ***bits_per_pixel*** * 2, respectively. However, the number of bits needed to code the final group can be similar to the number of bits required to code a group of 3 pixels, since partial groups are coded in the same fashion as 3-pixel groups except that unused pixels are discarded. Therefore, the extra unbudgeted bits must be made up for within the bits budgeted for the 3-pixel groups in the rest of the line. If the slice width is narrow, there are not many groups over which to make up the unbudgeted bits, particularly when ***bits_per_pixel*** is high.

Therefore, the warning code was changed to account for unbudgeted bits resulting from partial groups when performing the ***rc_range_parameters[14].range_max_qp*** check. This change does not affect any cases where the slice width is an integer multiple of 3 (or 6 if ***natve_422*** or ***native_420*** is used). Many typical DSC use cases involve large slice widths and thus would not be impacted by this change.

In order to help users make appropriate adjustments easily, GENERATE_RC_PARAMETERS can now be configured to a value of 2 to automatically adjust the ***rc_range_parameters[14].range_max_qp*** parameter if needed.

Background emails:

From: Aoun, Mike (PERATON)
Sent: Friday, September 17, 2021 7:15 PM
To: Frederick Walls
Subject: Possible bug?

Hi Fred,

I may have found a MINOR bug in the DSC program. It looks like during the encoding operation the SRC_LIST can take fully qualified path names (and in fact the readme file says "The list file may contain full pathnames"). However, it seems that when using the program to decode (FUNCTION 2), the program uses

.\<fname> instead of the fully qualified path that was specified. I stepped through the code and confirmed that there is a line that uses fn_o (the output directory, which by default is .\ ) concatenated with the base filename for the input file. Consequently, this strips off the "fully qualified" part of the input file path.

I thought I might be able to work around this by specifying the base filename in the SRC_LIST and using OUT_DIR to specify the rest of the path however, I found that spaces in the OUT_DIR path mess things up and double quotes don't help.

I understand that you wrote this to work in general with the test images and that in a practical sense everything works as needed by the typical user. I hope I am not being a nuisance by bringing this to your attention. Thanks again.

Regards,

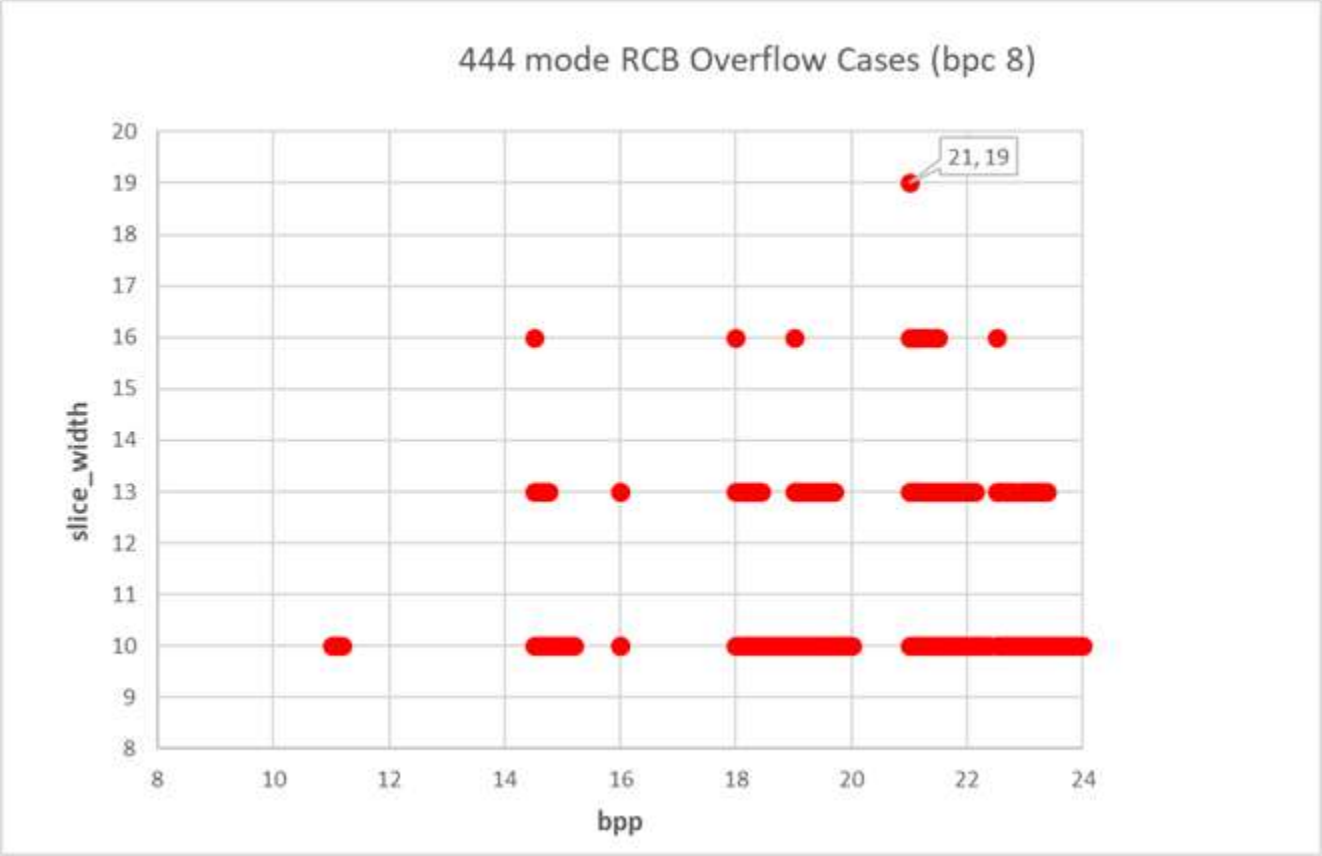
Mike

From: Fred Zhu
Sent: Monday, October 18, 2021 11:12 AM
To: Frederick Walls
Subject: RE: about DSC RCB overflow

Hi Fred,

I see, I also tried 444 mode, it also has similar issue,

e.g. slice_width =19, bpp=21 will overflow. Attach the files.



From: Fred Zhu <fzhu@nvidia.com>
Sent: Sunday, October 17, 2021 12:58 PM
To: Frederick Walls <frederick.walls@broadcom.com>
Subject: RE: about DSC RCB overflow

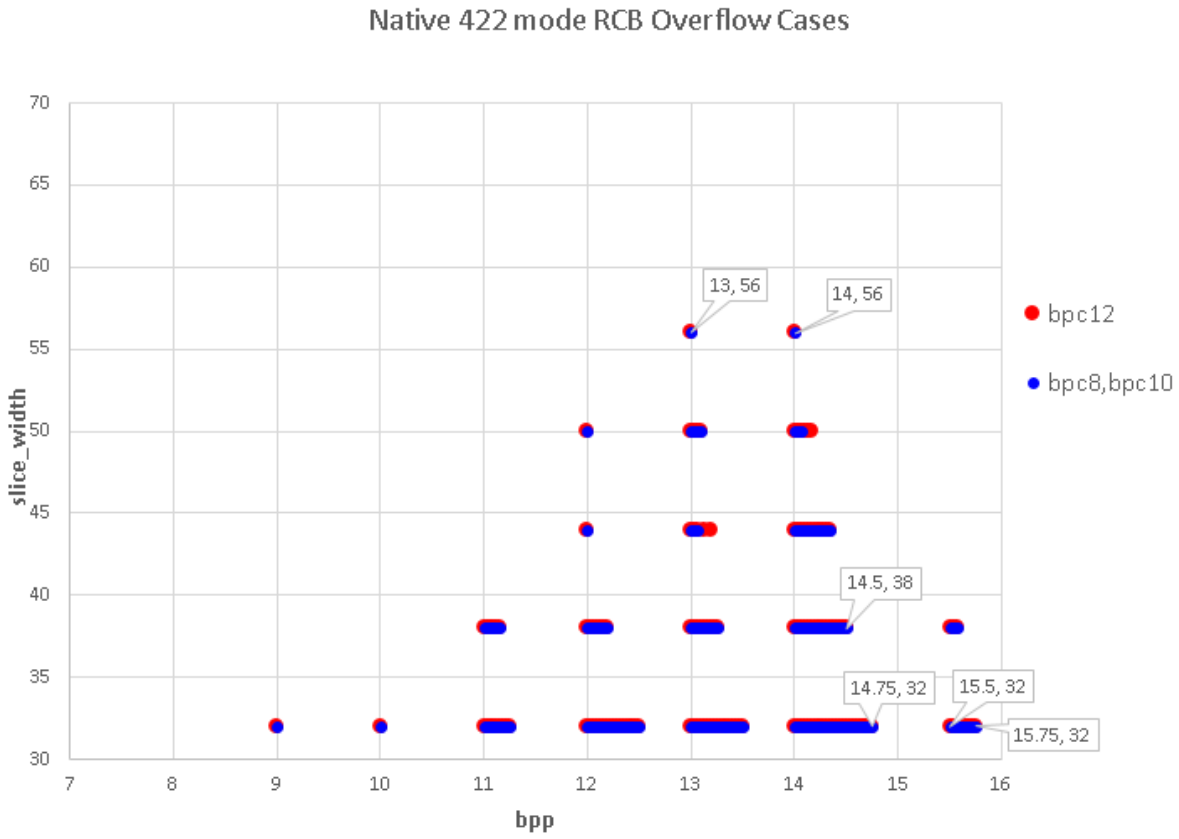
Hi Fred,

Yes, I agree, in real products, we won't use such small slice_width value, also random pixels are rare in real world.

There's no min slice_width specified in the spec, but I believe there must be one. I think it's better to have clarification or guidance about this.

I fixed 8bpc and 10bpc input issue as you pointed out and reran them all, all the failure cases are listed in the xls, and plotted chart like below.

The results of 8 bpc and 10bpc are exactly same, results of 12bpc have 5 more overflow cases at slice_width=44 and 50.



Thanks.

From: Fred Zhu
Sent: Wednesday, October 13, 2021 6:06 AM
To: Frederick Walls
Subject: about DSC RCB overflow

Hi Fred,

We are using DSC model to encode random images, we've found "RCB overflow" issue in some small slice_width cases.

As an example, in native 422 mode, we traversed all the possible cases like below:

bpc= 8/10/12, bpp from 7.0 ~ 16.0 with step 1/16, slice_width from 32 to 128 with step 2, slice_height=CEIL(15000/slice_width).

All the RCB overflow cases are plotted in the following diagrams.

I know that in real use cases, slice_width won't be very small, but for debug and test purpose, using small slice_width helps to reduce runtime.

My question is:

1, is (or will) there any guideline about the slice_width selection?

2, is (or will) there any guideline about RCB parameter adjustment in overflow cases?

Thanks.

- End of Document -