

Image Interleaver Pattern

The interleaver pattern class (**SvtMipiImageInterleaverPattern**) enables the interleaving of different image patterns. This includes datatype interleaving, burst interleaving, virtual channel (VC) interleaving and combinations of these.

The following sections assumes that the patterns are declared in numerical order in the **patterns** attribute, e.g.: **imageInterleaver.patterns = [pattern0, pattern1, pattern2, ...]**. Any reference to pattern0 refer to the first pattern in the pattern list, references to pattern1 to the second pattern, and so on.

Note that the image interleaver pattern operates the same way for both C-Phy and D-Phy platforms.

The interleaver pattern accepts as input any pattern of the following classes:

SvtMipiCsiImagePattern

SvtMipiDsiImagePattern

SvtMipiImageInterleaverPattern

All input patterns must be of the same protocol (CSI or DSI). This class supports nested interleaving; see section *Nested interleaving* for more information about using **SvtMipiImageInterleaverPattern** as input.

CSI interleaving modes

The CSI interleaver supports three modes of interleaving: interleaving by frames, by lines, or by virtual channel.

Frame interleaving (byFrames)

Frame mode interleaves the frames of each pattern following their input order in **patterns**. The frames can differ in size, resolution, line blanking, frame blanking, datatype, and virtual channel. However, note that the CSI specifications require having the same lincetime across frames [1, p.84]. A visual example can be found in Fig. 1.

For frame interleaving, pattern0 must have the the largest number of frames, with respect to other patterns in the pattern list.

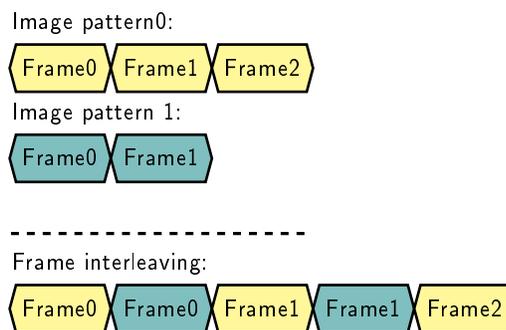


Figure 1: Example of frame interleaving for CSI. Each frame block contains all the parts pertaining to a single frame, including a pair of framestart/framend packets, data lines, frame blanking.

Line interleaving (byLines)

Line mode interleaves the lines of multiple frames within a single frame. The lines of each pattern are interleaved one after the other in the corresponding frame. The main use case of this interleaving method is to have multiple datatypes within a single frame. The frame blanking information, frame start (FS) and frame

end (FE) packets are taken from pattern0, and are removed from the other patterns prior to interleaving. Note that line interleaving is restricted to patterns operating on the same virtual channel. This interleaver supports interleaving lines with different linetimes, but keep in mind that the CSI specifications require having the same linetime within one frame [1, p.84]. A visual example can be found in Fig. 2.

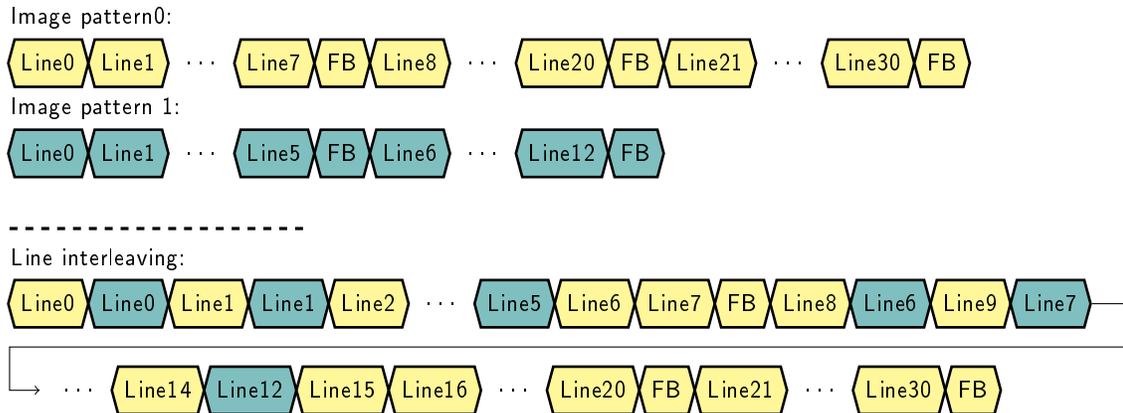


Figure 2: Example line interleaving for CSI. The FB blocks represent the frame blanking lines, and also represent the end of a frame. Each line block contains all parts pertaining to a single line, including SoT/EoT pairs, data, and line blanking. Note that the FS and FE packets get stripped off the concerned lines of pattern 1 while maintaining the same line linetime. The frame blanking used in this mode corresponds to the frame blanking of pattern0.

For line interleaving, pattern0 must have the most frames compared to other patterns, and each of those frames must have more lines than the corresponding frame in the other patterns.

Virtual channel interleaving (byVirtualChannel)

Virtual channel (VC) mode interleaves the lines of the input patterns, irrespective of the frame start/end points. Consequently, each channel may have different framerates. Unlike the two previous interleaving methods, there are no restrictions in the CSI specifications for the linetime of the different patterns. A visual example can be found in Fig. 3.

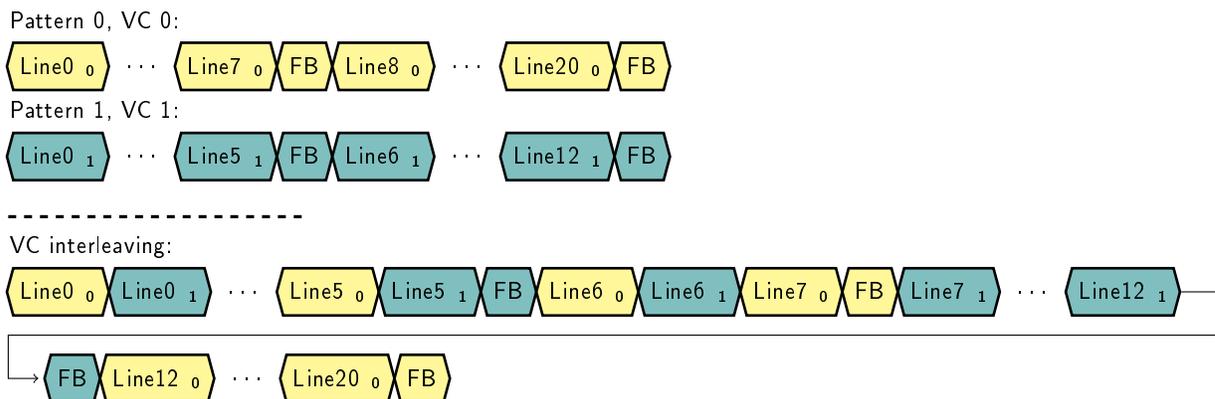


Figure 3: Example virtual channel interleaving for CSI. The FB blocks represent the frame blanking lines, and also represent the end of a frame. The underscore on the blocks represent the virtual channel of the packets contained in the line line contain. Each line block contains all parts pertaining to a single line, including SoT/EoT pairs, data, line blanking, and FS/FE packets where needed.

For VC interleaving, pattern0 must have more total lines (sum of lines across all frames) than the other input patterns.

DSI Interleaving modes

The DSI interleaver supports two modes: interleaving by packets or by bursts. Contrasting with CSI interleaving, there are no restriction regarding what constitutes valid input. Any datatype or virtual channels can be interleaved together. Both modes are depicted in Fig. 4.

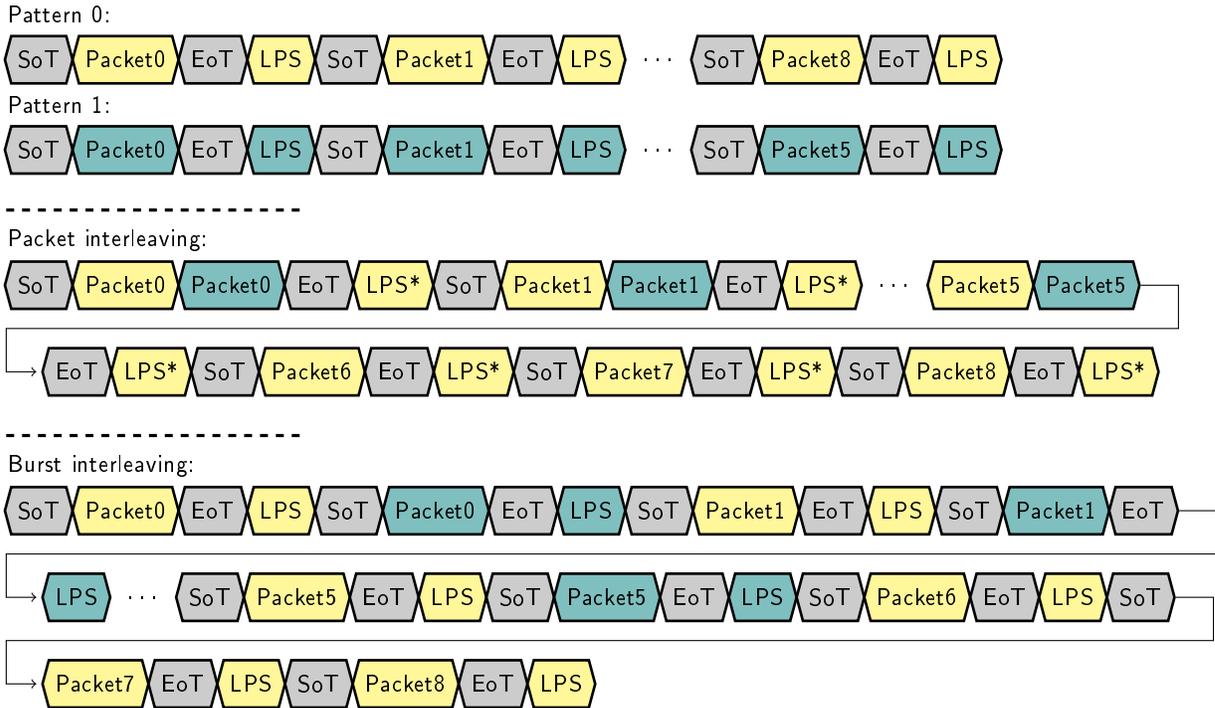


Figure 4: DSI interleaving modes. The LPS block corresponds to a low-power stop. For Packet interleaving, note that LPS* corresponds to the adjusted stop blocks, such that pattern0's linetime is maintained.

Packet interleaving (byPackets)

Packet interleaving merges the concurrent bursts of each pattern. More specifically, the packets of the n^{th} burst of patterns 1,2,... are appended to the packets of the n^{th} burst of pattern0. The line blanking is adjusted after each insertion, such that the linetime of the final interleaved pattern matches the linetime of pattern0.

For packet interleaving, pattern0 must have more bursts than the other patterns in the pattern list.

Burst interleaving (byBursts)

Burst interleaving instead appends the bursts one after another. The line blanking of each pattern is carried over to the final interleaved pattern.

For burst interleaving, pattern0 must have more bursts than the other patterns in the pattern list.

patternRepeats parameter

The **patternRepeats** attribute acts as a multiplier to the corresponding entry in **patterns**. For example, in Fig. 1, the final pattern starts and ends with a frame from pattern0, which may be an undesired behaviour. This situation can be remedied with appropriate use of the **patternRepeats** list:

```
imageInterleaver.patterns = [pattern0, pattern1]
```

`imageInterleaver.patternRepeats = [2, 3]`

This is visually depicted in Fig. 5.

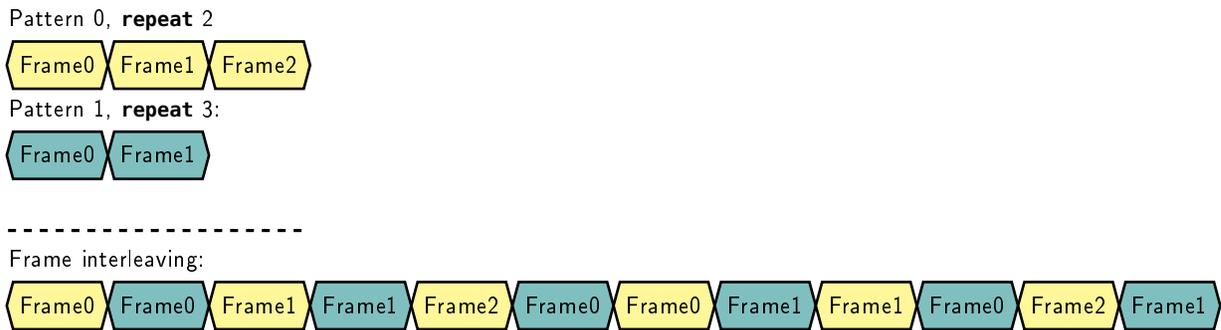
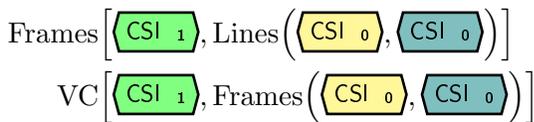


Figure 5: Example of frame interleaving for CSI with different repeat value for each input pattern.

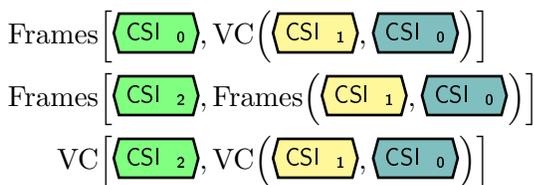
Another use of the **patternRepeats** parameter is in the case where pattern0 is too short for the interleaving mode. In such a case, simply increase pattern0's **patternRepeats** until it is long enough.

Nested interleaving

Each interleaver pattern can be re-interleaved into another interleaver to produce more complex results. For DSI, any interleaver pattern is valid input for another DSI interleaver. For CSI, there is a restriction where an input pattern cannot contain multiple virtual channels. For example, the following are valid interleaving chains:



where the subscript on the blocks is the virtual channel, and the VC, Frames and Lines are the interleaving modes. The first interleaving chain starts by interleaving the *Lines* of CSI 0 and CSI 0, then interleaves the output's frames with with CSI 1; see the section *CSI interleaving modes* for more details. On the other hand, the following interleaving chains are disallowed for CSI, as the input to the outermost interleavers contain multiple virtual channels:



References

[1] Mipi Alliance, "Specification for camera serial interface 2 (csi-2)," oct 2014. version 1.3.