Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: October 29, 2021

J. Weaver
BBC R&D

April 27, 2021

Packetised Essence Format for Uncompressed Video
draft-weaver-pef-00

Abstract

This memo specifies a new proposed packing format for Uncompressed video data at a variety of bit-depths and with a variety of different component structures.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

This memo specifies a format for packing uncompressed video of a variety of formats into an easily processed format for software. It is compatible with video of a variety of bit-depths, colour subsampling formats, picture resolutions, frame rates, scan-modes, and number of components.

1.1. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

2. Video Data Formats, an Overview

For purposes of this recommendation a video format is defined by a number of parameters:

- The components of the video. See the fuller discussion below for what this means in general.

- The number of samples across a line of video, which may vary for different components (for example for video with a colour subsampling other than 4:4:4).

- The number of lines in a frame of video, which may vary for different components.

- The bit-depth of the samples. Currently supported bit-depths are 1, 2, 4, 8, 9, 10, 12, 14, and 16. This can also vary for different components. For example an Alpha channel or a depth map may use a lower bit depth than the main luma channel

- The frame-rate of the video.

For the purposes of this format we support an arbitrary number of video components. A video component is a plane of samples for the video which is in some way independent of other samples outside of that plane. For progressively scanned monochrome video there is only one component, which is the video data's brightness (luma) values. For monochrome video with an interlaced scan mode, on the other hand, there are two components per frame: the luma of the first field and the luma of the second field. For colour video there are generally three components per field, either Luma, and two colour-difference signals (Y,Cb,Cr) or three colour channels (R,G,B). In this case for progressively scanned video there will be three components, eg. Y, Cb, and Cr, and for video which is interlaced there would be six, for example Field 1 Y, Field 1 Cb, Field 1 Cr, Field 2 Y, Field 2 Cb, and Field 2 Cr. Other video formats may have different numbers or arangements of components: for example a stereoscopic, interlaced, video stream making use of three colour channels and an alpha channel might have sixteen components: Left Field 1 Y, Left Field 1 Cb, Left Field 1 Cr, Left Field 1 A, Right Field 1 Y, Right Field 1 Cb, Right Field 1 Cr, Right Field 1 A, Left Field 2 Y, Left Field 2 Cb, Left Field 2 Cr, Left Field 2 A, Right Field 2 Y, Right Field 2 Cb, Right Field 2 Cr, and Right Field 2 A. Future video formats may contain additional components which have yet to considered, but should still be compatible with this format.

There are a plethora of existing video formats which arrange the video in a variety of ways. Some formats are "interleaved" -- ie. they arrange samples from different components one after another in order. Other formats are "planar" -- they arrange the data so that all the samples for one component for a frame or field are placed together and followed by all the samples for another component for the same frame or field. In addition the arrangement of the data samples themselves within memory is also variable: some formats are called "packed" -- meaning that samples with a bit-depth which is not a power of two are placed within memory in a bit-contiguous fashion regardless of octet boundaries; others are "padded", meaning that samples are placed in the low-order bits of a larger octet-aligned container; Finally there are formats which both pad and pack the samples, the most well-known being V210, a format which places three 10-bit samples into the least significant bits of a 32-bit word, but ignores the value of the top two bits.

The advantage of "packed" formats is that they make use of all the bits available for storing video data, thus making them more space efficient in storage or more bandwidth efficient when used on a network. The disadvantage is that they are complicated to deal with in software, since computer architectures are generally optimised to process multiples of 8, 16, 32, or 64 bits.

The format recommended here uses a different approach. Rather than storing data samples interleaved or in a planar structure this format arranges them into "blocks" of samples, which are then stored in a planar fashion. The size of these "blocks" of samples can be selected so as to correspond to the block sizes used in various video compression and video processing techniques, allowing a more natural processing when using these techniques, or can be adjusted to provide a lower latency as required by the specific application in use.

Rather than traditional packing or padding this format proposes the storage of the bits in the samples in a non-contiguous structure, allowing for 100% utilisation of the available bits, like a packed format, but with all data aligned conveniently on power-of-two boundaries as found in a padded format.

Packed 10-bit:

```
   0                   1                   2                   3
   0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |     Sample 3      |     Sample 2      |     Sample 1      |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    7      |     Sample 6      |     Sample 5      |     Sample 4
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   mple 10     |     Sample 9      |     Sample 8      |     Sample
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     Sample 13     |     Sample 12     |     Sample 11     |     Sam
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |     Sample 16     |     Sample 15     |     Sample 14     |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Padded 10-bit in 16-bit:

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |X X X X X X|     Sample 1      |X X X X X X|     Sample 2      |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |X X X X X X|     Sample 3      |X X X X X X|     Sample 4      |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |X X X X X X|     Sample 5      |X X X X X X|     Sample 6      |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

V210 Format (a hybrid of packing and padding):

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |X X|     Sample 3      |     Sample 2      |     Sample 1      |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |X X|     Sample 6      |     Sample 5      |     Sample 4      |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |X X|     Sample 9      |     Sample 8      |     Sample 7      |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 1: Examples of packing formats

3. The Proposed Format in Detail

When making use of this format a number of parameters must be set. These will be introduced in the text below as they are explained. For the purposes of this format the components of the video are treated separately. Some parameters must be the same for all components, but others may vary, and this will be noted below. Fundamentally this format involves breaking the video samples in a component for a specific frame into "blocks", which is to say two dimensional groupings of samples which form a rectangular shape in the active part of the frame.

Bit Depth (D): This is the number of bits used in each sample. In principle this may vary by component, but in general it will usually be the same for all components. Currently supported bit-depths are: 1, 2, 4, 8, 9, 10, 12, 14, and 16.

Block Height (BH) and Block Width (BW): These are integer values which give a number of samples of height and width for each block. There is no requirement for all components to use the same values for these, but there are restrictions on the valid choices of BH and BW.

```
   0                   1
   0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9
00 X X X X X X X X Y Y Y Y Y Y Y Y
 1 X X X X X X X X Y Y Y Y Y Y Y Y
 2 X X X X X X X X Y Y Y Y Y Y Y Y
 3 X X X X X X X X Y Y Y Y Y Y Y Y
 4 X X X X X X X X Y Y Y Y Y Y Y Y
 5 X X X X X X X X Y Y Y Y Y Y Y Y
 6 X X X X X X X X Y Y Y Y Y Y Y Y
 7 X X X X X X X X Y Y Y Y Y Y Y Y
 8 Z Z Z Z Z Z Z Z
 9 Z Z Z Z Z Z Z Z
10 Z Z Z Z Z Z Z Z
 1 Z Z Z Z Z Z Z Z
 2 Z Z Z Z Z Z Z Z
 3 Z Z Z Z Z Z Z Z
 4 Z Z Z Z Z Z Z Z
 5 Z Z Z Z Z Z Z Z
 6
 7
 8
 9
```

Figure 2: Example of Blocks (X marks one 8x8 block, Y another, Z a third)

D	Restriction on BH and BW
1	BW x BH MUST be a multiple of 8
2	BW x BH MUST be a multiple of 4
4	BW x BH MUST be a multiple of 2
8	No restriction
9	BW x BH MUST be a multiple of 128
10	BW x BH MUST be a multiple of 64
12	BW x BH MUST be a multiple of 32
14	BW x BH MUST be a multiple of 64
16	No restriction

Each frame of video MUST be represented by one stream of octets per component. The ordering of the components is a parameter of the video format itself. So a frame of YCbCr video will consist of a stream of luma, followed by two streams of chroma differences, whilst a frame of RGBA video will consist of a stream of Red, one of Green, one of Blue, and one of Alpha.

Each stream is assembled starting by taking the blocks for that component in that frame in raster-scan order. If the width of a block is not a divisor of the number of samples across an active line of video for that component, or if the height is not a divisor of the height of the active video for that component then the data MUST be padded with additional samples to fill the block size. The choice of values for padding is not specified, and MAY be any value desired. Mirroring, edge-extending, padding with black, and padding with mid-grey are all possible options.

Each block of samples is converted into a stream of octets as follows:

- If D=1 simply pack the values of the samples in raster scan order as 1-bit values into octets. Each block will thus be represented by BW x BH / 8 octets.

- If D=2 simply pack the values of the samples in raster scan order as 2-bit values into octets. Each block will thus be represented by BW x BH / 4 octets.

- If D=4 simply pack the values of the samples in raster scan order as 4-bit values into octets. Each block will thus be represented by BW x BH / 2 octets.

- If D=8 simply use the values of the samples in raster-scan order with each sample value occupying one octet. Each block will thus be represented by BH x BW octets.

- If D=9 pack the least significant bit of each sample in raster scan order into the first BH x BW x 0.125 octets, then use the most significant 8 bits of each sample in the same order as the values of the next BH x BW octets. Each block will thus be represented by BH x BW x 1.125 octets. This arrangement, together with the restrictions on the choice of BH and BW ensures that the number of octets used to store the least significant bits will always be a multiple of 16.

- If D=10 pack the least significant 2 bits of each sample in raster scan order into the first BH x BW x 0.25 octets, then use the most significant 8 bits of each sample in the same order as the values of the next BH x BW octets. Each block will thus be represented by BH x BW x 1.25 octets. This arrangement, together with the restrictions on the choice of BH and BW ensures that the number of octets used to store the least significant bits will always be a multiple of 16.

- If D=12 pack the least significant 4 bits of each sample in raster scan order into the first BH x BW x 0.5 octets, then use the most significant 8 bits of each sample in the same order as the values of the next BH x BW octets. Each block will thus be represented by BH x BW x 1.5 octets. This arrangement, together with the restrictions on the choice of BH and BW ensures that the number of octets used to store the least significant bits will always be a multiple of 16.

- If D=14 pack the least significant 2 bits of each sample in raster scan order into the first BH x BW x 0.25 octets, then pack the next least significant 4 bits of each sample in the same order into the next BH x BW x 0.5 octets, then use the most significant 8 bits of each sample in the same order as the values of the next BH x BW octets. Each block will thus be represented by BH x BW x 1.75 octets. This arrangement, together with the restrictions on the choice of BH and BW ensures that the number of octets used to store the least significant bits will always be a multiple of 16,

as will the number of octets used to store the next least significant bits of each sample.

o If D=16 use the values of the samples in raster-scan order with each sample value occupying two octets in network octet order. Each block will thus be represented by BH x BW x 2 octets.

As an example of how the data is structured refer to Figure 3, which provides an example of a 64-sample block (such as an 8x8 block). The initial section consist of 16 octets containing packed 2-bit values each containing bits 8 and 9 of each sample, whilst the remaining 64 octets contain 8-bit values containing bits 0 to 7 of each sample.

The more complex 14-bit packing is shown in Figure 4. In this case the first 16 octets contain packed 2-bit values containing bits 12 and 13 of each sample, the next 32 octets contain packed 4-bit values containing bits 8-11 of each sample, and the remaining 64- octets contain 8-bit values consisting of bits 0 to 7 of each sample.

Following these rules each component of a frame of 1920 x 1080 10-bit progressive video with a block shape of 8x8 for all components will contain 32400 blocks, each represented by 80 octets broken into a 16-octet first part containing the least significant 2 bits of each sample and a 64-octet second part. In this fashion this packing format is bit-rate efficient, making full use of all available bits.

```
  0                   1                   2                   3
  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
L+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
S| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |10 |11 |12 |13 |14 |15 |16 |
B+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
s|17 |18 |19 |20 |21 |22 |23 |24 |25 |26 |27 |28 |29 |30 |31 |32 |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |33 |34 |35 |36 |37 |38 |39 |40 |41 |42 |43 |44 |45 |46 |47 |48 |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |49 |50 |51 |52 |53 |54 |55 |56 |57 |58 |59 |60 |61 |62 |63 |64 |
M+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
S|  Sample 1     |  Sample 2     |  Sample 3     |  Sample 4     |
B+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
s|  Sample 5     |  Sample 6     |  Sample 7     |  Sample 8     |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                                ...
 |  Sample 61    |  Sample 62    |  Sample 63    |  Sample 64    |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 3: Arrangement of Data for an 8x8 block when D=10

```
     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   L+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   S| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |10 |11 |12 |13 |14 |15 |16 |
   B+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   s|17 |18 |19 |20 |21 |22 |23 |24 |25 |26 |27 |28 |29 |30 |31 |32 |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |33 |34 |35 |36 |37 |38 |39 |40 |41 |42 |43 |44 |45 |46 |47 |48 |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |49 |50 |51 |52 |53 |54 |55 |56 |57 |58 |59 |60 |61 |62 |63 |64 |
   M+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   i|   1   |   2   |   3   |   4   |   5   |   6   |   7   |   8   |
   d+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   d|   9   |  10   |  11   |  12   |  13   |  14   |  15   |  16   |
   l+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   e                               ...
    |  57   |  58   |  59   |  60   |  61   |  62   |  63   |  64   |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   M|  Sample 1     |  Sample 2     |  Sample 3     |  Sample 4     |
   S+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   B|  Sample 5     |  Sample 6     |  Sample 7     |  Sample 8     |
   s+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                                   ...
    |  Sample 61    |  Sample 62    |  Sample 63    |  Sample 64    |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 4: Arrangement of Data for an 8x8 block when D=14

4. Security Considerations

A blocking format for uncompressed video does not in of itself have any particular security implications, but also does not provide any additional security. If the content is sensitive then some other mechanism needs to be used to keep it secure.

5. IANA Considerations

This format would require a mime type and associated RTP payload type to be registered.

5.1. Proposed MIME Type Registration

Type name: video

Subtype name: pef

Required parameters:

sampling: Determines the color (sub-)sampling mode of the video stream. Currently defined values are:

Monochrome: (in which case for progressive scan the only component is component 0),

RGB: (in which case for progressive scan component 0 is R, component 1 is G, and component 2 is B),

RGBA: (in which case for progressive scan component 0 is R, component 1 is G, component 2 is B, and component 3 is A),

BGR: (in which case for progressive scan component 0 is B,component 1 is G, and component 2 is R),

BGRA: (in which case for progressive scan component 0 is B, component 1 is G, component 2 is R, and component 3 is A),

YCbCr: (in which case for progressive scan component 0 is Y, component 1 is Cb, and component 2 is Cr),

New values may be registered as needed.

width: A comma separated list of values which determines the number of samples per line for each component. Each value is an integer.

height: A comma separated list of values which determines the number of lines per frame for each component. Each value is an integer.

depth: A comma separated list of values giving the number of bits per sample. Each value is an integer with currently permissible values of 1, 2, 4, 8, 9, 10, 12, 14, and 16.

colorimetry: This parameter defines the set of colorimetric specifications and other transfer characteristics for the video source, by reference to an external specification. Valid values and their specification are:

BT601-5: ITU Recommendation BT.601-5

BT709-2: ITU Recommendation BT.709-2

SMPTE240M: SMPTE standard 240M

block-width: A comma separated list of integers, each is the width of each block in samples for a specific component.

block-height: A comma separated list of integers, each is the width of each block in samples for a specific component.

Optional parameters:

interlace: If this OPTIONAL parameter is present, it indicates that the video stream is presented as interlaced. If absent, progressive scan is implied. If present then the number of components in the stream is double that found in a progressively scanned stream with the same sampling parameter.

stereo: If this OPTIONAL parameter is present, it indicates that the video stream is presented as a stereo pair. If present then the number of components in the stream is double that found in a stream with the same sampling parameter.

chroma-position: This OPTIONAL parameter defines the position of chrominance samples relative to luminance samples. It is either a single integer or a comma separated pair of integers. Integer values range from 0 to 8, as specified in Figures 6-8 of [RFC4175]. A single integer implies that Cb and Cr are co-sited. A comma separated pair of integers designates the locations of Cb samples, respectively. In its absence, a single value of assumed for color-subsampled video (chroma-position=0).

gamma: An OPTIONAL floating point gamma correction value.

Encoding considerations: framed

Security considerations: This format contains no active content, and has no particular security considerations.

Interoperability considerations: This format is an alternative form of raw uncompressed video distinct from and incompatible with video/raw.

Published specification: This Document

Applications that use this media type: Video communication.

Fragment identifier considerations: N/A

Additional information:

File Name Extensions: .pef

Person & email address to contact for further information: James P. Weaver < james.barrett@bbc.co.uk >

Intended usage: COMMON

Restrictions on usage: N/A

Author: James P. Weaver

Change controller: James P. Weaver

Provisional registration? (standards tree only): YES

6. Unpacking of Video Data in Software and Hardware

The format here described has been designed with an eye to ease of unpacking in software and hardware. In the case of 8-bit video this format is very similar to the conventional 8-bit formats and so requires no additional processing. A traditional 10-bit format, however, either packs 10-bit samples in a way which overlaps octet boundaries or pads, and thereby doesn't make use of all bits. This format makes full use of all bits like a packed format, but maintains octet alignment.

For software processing purposes it is common to unpack 10-bit data so that each sample is stored in the lower 10-bits of a 16-bit word, essentially converting from a packed format to a padded one. To do this for the traditional packed 10-bit format is a complex operation involving a significant amount of masking and shifting and is not easily amendable to processing using the efficient 128-bit Single Instruction Multiple Data instructions implemented on many modern processors.

By comparison this new format is easily convertible to a padded format using SIMD instructions, with a general algorithm as described in the following. This algorithm has been kept general and does not refer to specific SIMD instructions since the likelihood is that instruction sets will change in the future.

The exact mechanism used for the decode will depend upon the size of the registers on which the SIMD instructions being used operate. For these examples I will explain with 4 octet registers, simply because it makes the diagrams more easily viewed.

Starting with an 80-octet stream containing the data representing a 64 sample block of 10-bit video data we begin by loading the first four octets of the block into a register:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |10 |11 |12 |13 |14 |15 |16 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

(The numbers in the centre of each two-bit space indicate the sample number from which that data originates)

Then using a masked move or shuffle we are able to "spread" this data out into the octets of two registers:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|X X X X X X X X| 1 | 2 | 3 | 4 |X X X X X X X X| 1 | 2 | 3 | 4 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|X X X X X X X X| 1 | 2 | 3 | 4 |X X X X X X X X| 1 | 2 | 3 | 4 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Then using a SIMD multiplication instruction which treats each 32-bit register as two 16-bit values we multiply each block up in order to align the required data at the top of the low order octet of each 16-bit sample:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|X X X X X X X X| 1 | 2 | 3 | 4 |X X X X X X| 1 | 2 | 3 | 4 |X X|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|X X X X| 1 | 2 | 3 | 4 |X X X X X X| 1 | 2 | 3 | 4 |X X X X X X|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Then the data from the first four octets of the second segment of the data stream (where the MSBs of the samples are stored) are loaded into a register

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Sample 1 MSBs | Sample 2 MSBs | Sample 3 MSBs | Sample 4 MSBs |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

and a shuffle and move or blanking and bitwise or is used to distribute these octets into the high order octets of each 16-bit word in the working registers.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Sample 1 MSBs | 1 | 2 | 3 | 4 | Sample 2 MSBs | 2 | 3 | 4 |X X|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Sample 3 MSBs | 3 | 4 |X X X X| Sample 4 MSBs | 4 |X X X X X X|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Finally a SIMD shift is used to shift each 16-bit value down by 6 bits:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|X X X X X X| Sample 1 MSBs | 1 |X X X X X X| Sample 2 MSBs | 2 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|X X X X X X| Sample 3 MSBs | 3 |X X X X X X| Sample 4 MSBs | 4 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Completing the conversion of the first four samples of the block from this format to a padded 10-bit in 16-bit format more convenient for processing in software. The conversion can then be continued with samples 5,6,7, and 8.

A larger register length for the SIMD operations enables more samples to be decoded simultaneously. This algorithm has been used effectively with 128-bit (16 octet) registers.

7. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC4175] Gharai, L. and C. Perkins, "RTP Payload Format for Uncompressed Video", RFC 4175, DOI 10.17487/RFC4175, September 2005, <https://www.rfc-editor.org/info/rfc4175>.

Author's Address

James P. Weaver
BBC R&D
Floor 5
Dock House
MediaCity UK
Salford M50 2LH
United Kingdom

Email: james.barrett@bbc.co.uk