

IPU Stream Format

© 2002 Sony Computer Entertainment Inc.

Publication date: December 2002

Sony Computer Entertainment Inc.
1-1, Akasaka 7-chome, Minato-ku
Tokyo 107-0052, Japan

Sony Computer Entertainment America
919 E. Hillsdale Blvd.
Foster City, CA 94404, U.S.A.

Sony Computer Entertainment Europe
30 Golden Square
London W1F 9LD, U.K.

The *Stream Data Creation* manual is supplied pursuant to and subject to the terms of the Sony Computer Entertainment PlayStation® license agreements.

The *Stream Data Creation* manual is intended for distribution to and use by only Sony Computer Entertainment licensed Developers and Publishers in accordance with the PlayStation® license agreements.

Unauthorized reproduction, distribution, lending, rental or disclosure to any third party, in whole or in part, of this book is expressly prohibited by law and by the terms of the Sony Computer Entertainment PlayStation® license agreements.

Ownership of the physical property of the book is retained by and reserved by Sony Computer Entertainment. Alteration to or deletion, in whole or in part, of the book, its presentation, or its contents is prohibited.

The information in the *Stream Data Creation* manual is subject to change without notice. The content of this book is Confidential Information of Sony Computer Entertainment.

 and PlayStation are registered trademarks of Sony Computer Entertainment Inc. All other trademarks are property of their respective owners and/or their licensors.

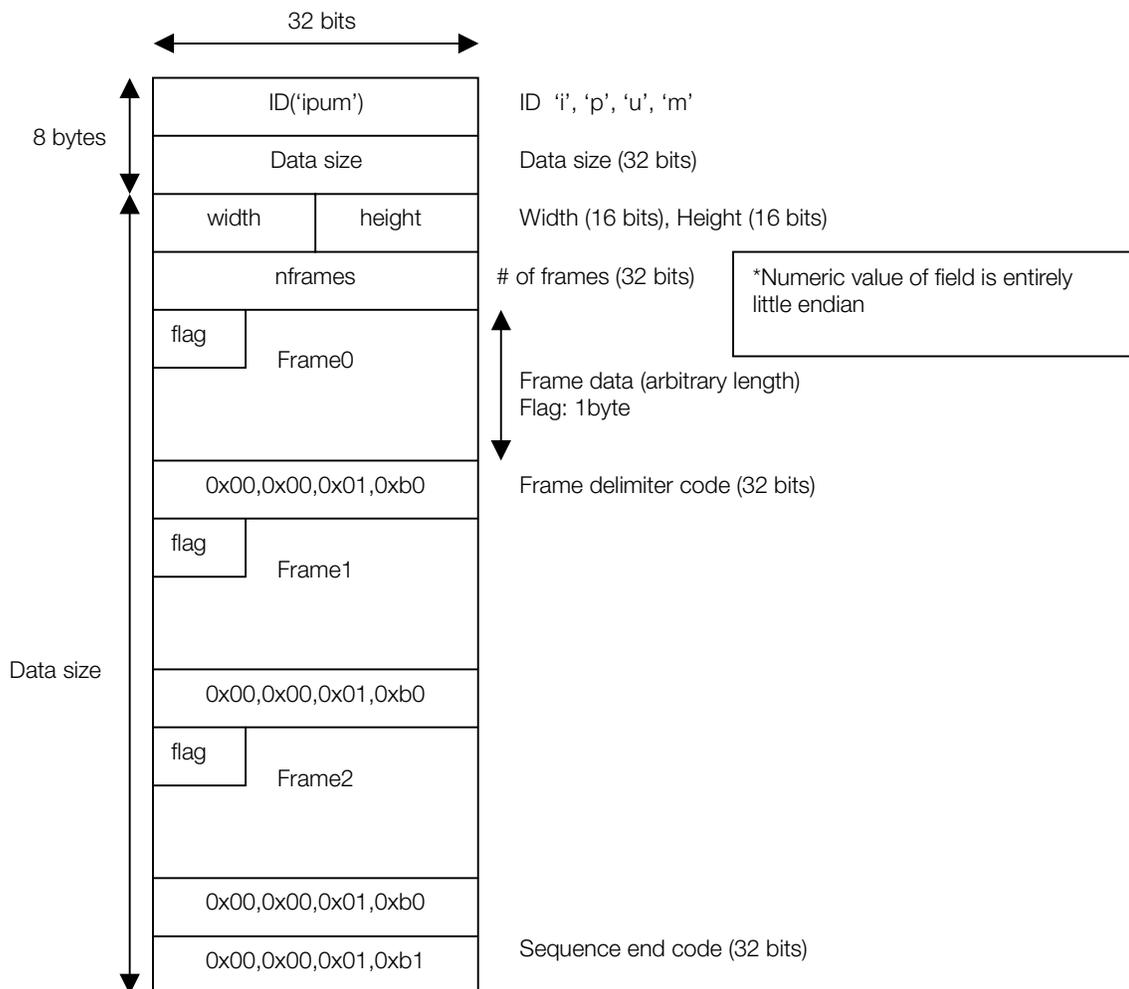
Table of Contents

Overview	5
Frame Data	5
Converting from MPEG2 Bit Stream	8
Changing the Structure of a Slice	8
Changing the DC Coefficient	8
Changing the Macroblock Type and Quantizer Scale Code	8
Flag Information	9

Overview

The IPU stream format is a video format that can be handled directly by the IPU. The IPU stream format provides picture quality and compression comparable to MPEG2 I-picture only streams. More data is required to provide picture quality comparable to standard MPEG2 streams with P-pictures and B-pictures, but there is less decoding overhead compared to MPEG2. Consequently, the IPU stream format is useful for performing decoding simultaneously with other operations. The structure of the IPU stream format is shown below.

Figure 1



Frame Data

In the IPU stream format, frames of data are separated by a four-byte delimiter code (0x00,0x00,0x01,0xb0). Individual frames are not mutually related, so each frame is independent. Frame data is converted from an MPEG2 I-Picture bit stream in such a way that it can be efficiently decoded by the IPU. The main difference between an IPU stream and an MPEG2 I-Picture bit stream is that in an IPU stream, the entire frame forms a large, continuous slice. Also, there are no slice-related attributes that are at the start of each slice as in MPEG2.

Flag information is placed at the start of frame data. Flag information takes up one byte, seven bits of which are used during decoding. Frame data flag information is shown below.

Figure 2

Flag Information

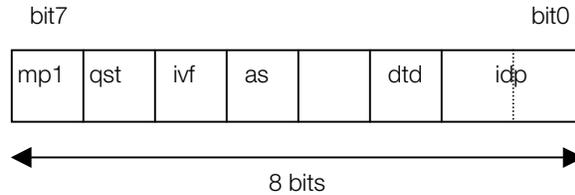


Table 1

Symbol	Meaning	Length
mp1	Mpeg1 bit stream 0: mpeg2 1: mpeg1	1 bit
qst	Q scale type 0: linear steps 1: non-linear steps	1 bit
ivf	Intra VLC format 0: mpeg1-compatible 2-dimensional VLC table 1: intra-macro block 2-dimensional VLC table	1 bit
as	Alternate scan 0: zig-zag scans 1: alternate scans	1 bit
dtd	DCT type decode 0: do not decode DCT type 1: decode DCT type	1 bit
idp	Intra DC precision 00: 8bit 01: 9bit 10: 10bit 11: reserved	2 bits

Frame data that follows flag information has the following format.

Figure 3

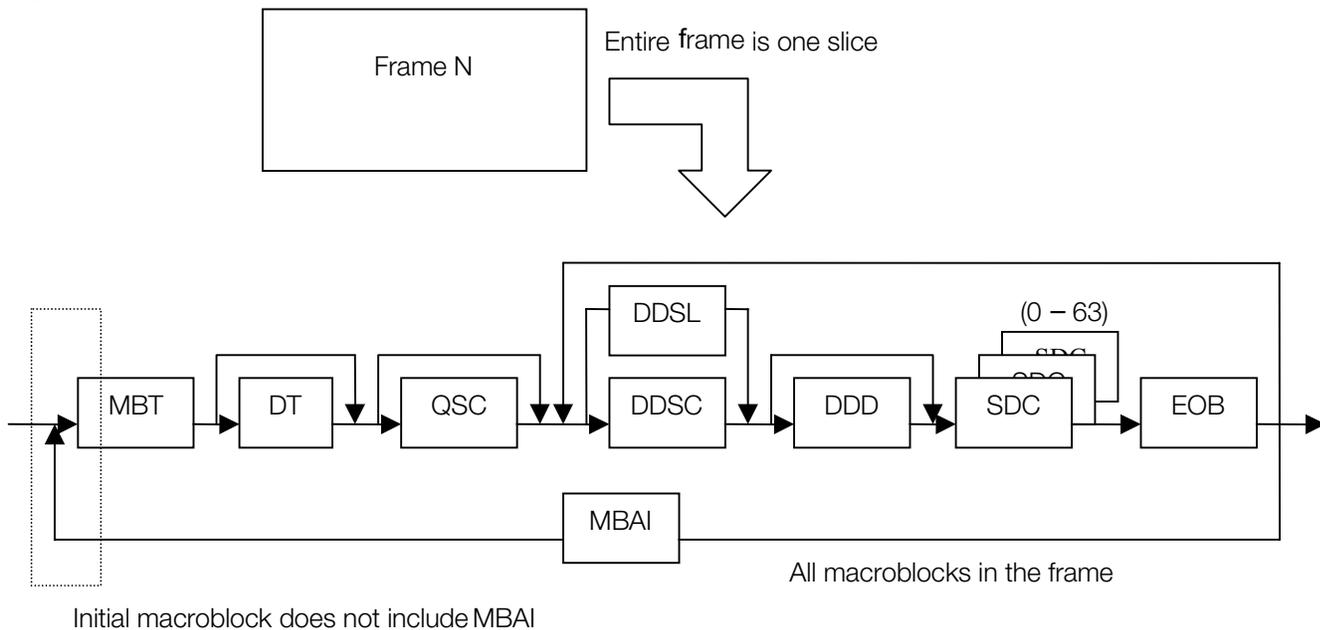


Table 2

Symbol	Name
MBAI	Macroblock Address Increment
MBT	Macroblock Type
DT	DCT Type
QSC	Quantiser Scale Code
DDSL	DCT DC Size Luminance
DDSC	DCT DC Size Chrominance
DDD	DCT DC Differential
SDC	Subsequent DCT Coefficients
EOB	End Of Block

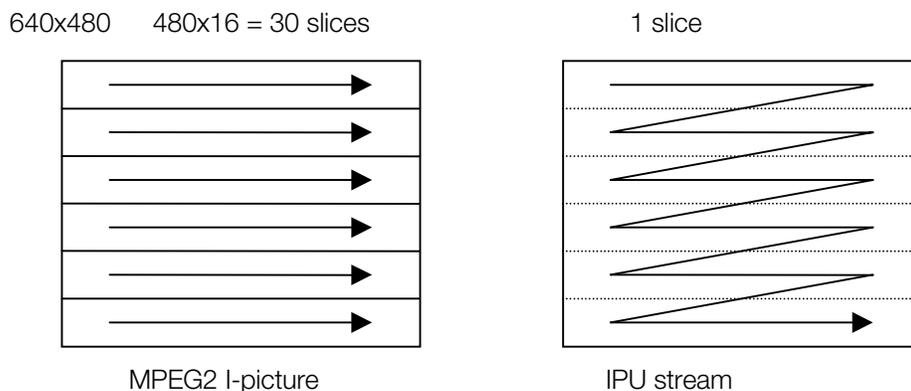
Converting from MPEG2 Bit Stream

IPU streams are created by converting MPEG2 streams consisting solely of I-pictures. The ps2str stream converter is generally used to perform this conversion. However, conversion can also be performed independently without a stream converter by using the method described below.

Changing the Structure of a Slice

In an MPEG2 bit stream, for example, a 640x480 image can be represented with 30 slices as shown below. However, in an IPU stream this would be represented as a single slice consisting of these 30 slices arranged continuously. In this manner, an IPU stream consists of a single slice for each image. This requires that the slice structure be changed when converting from an MPEG2 bit stream to an IPU stream.

Figure 4



Changing the DC Coefficient

When converting slices, the dc coefficient used for DCT must be changed. In MPEG2, the dc coefficient prediction value is reset at the start of each slice. However, the macroblock that was at the start of an MPEG2 stream slice is positioned in the middle of the slice in the IPU stream, preventing the dc coefficient from being reset. To convert data correctly for macroblocks that are at the start of an MPEG2 slice, the prediction value of the dc coefficient, which changes when it is reset, must be added to the original dc difference value.

Changing the Macroblock Type and Quantizer Scale Code

Another issue that must be considered with regard to changing slice structure is the quantizer scale code. In MPEG2, a quantizer scale code is always specified at the start of a slice. However, since the entire image is a single slice in an IPU stream, the MPEG2 quantizer scale code, which is specified per slice, cannot be used directly. To handle this, a macroblock at the start of an MPEG2 slice is converted into a macroblock accompanied by a quantizer scale code (macroblock_type=0x11). By indicating the quantizer scale code specified for the source slice, decoding can be performed in the same manner as the original MPEG2 bit stream.

- If a macroblock at the start of an MPEG2 slice does not have a quantizer scale code setting, it is converted to a macro block with a quantizer scale code specified at the start of the slice.

Flag Information

Flag information in frame data can be obtained from an MPEG2 bit stream in the following manner.

- mp1

mp1 is determined by whether or not there is an extension_start_code right after the first sequence_header in the bit stream. If there is an extension_start_code: mp1 = 0. If there is no extension_start_code: mp1 = 1.

- qst, ivf, as, idp

qst, ivf, as, idp indicate the flag information in picture_coding_extension.

- dtd

dtd is determined as follows:

```
dtd = (picture_structure == "Frame picture")
      && (frame_pred_frame_dct == 0)
```

